PREMIERE ISSUE

January 1983
USA $2.95 (UK£2.00)

inCider™

Green's Apple* Magazine

Applesoft Hints & Kinks

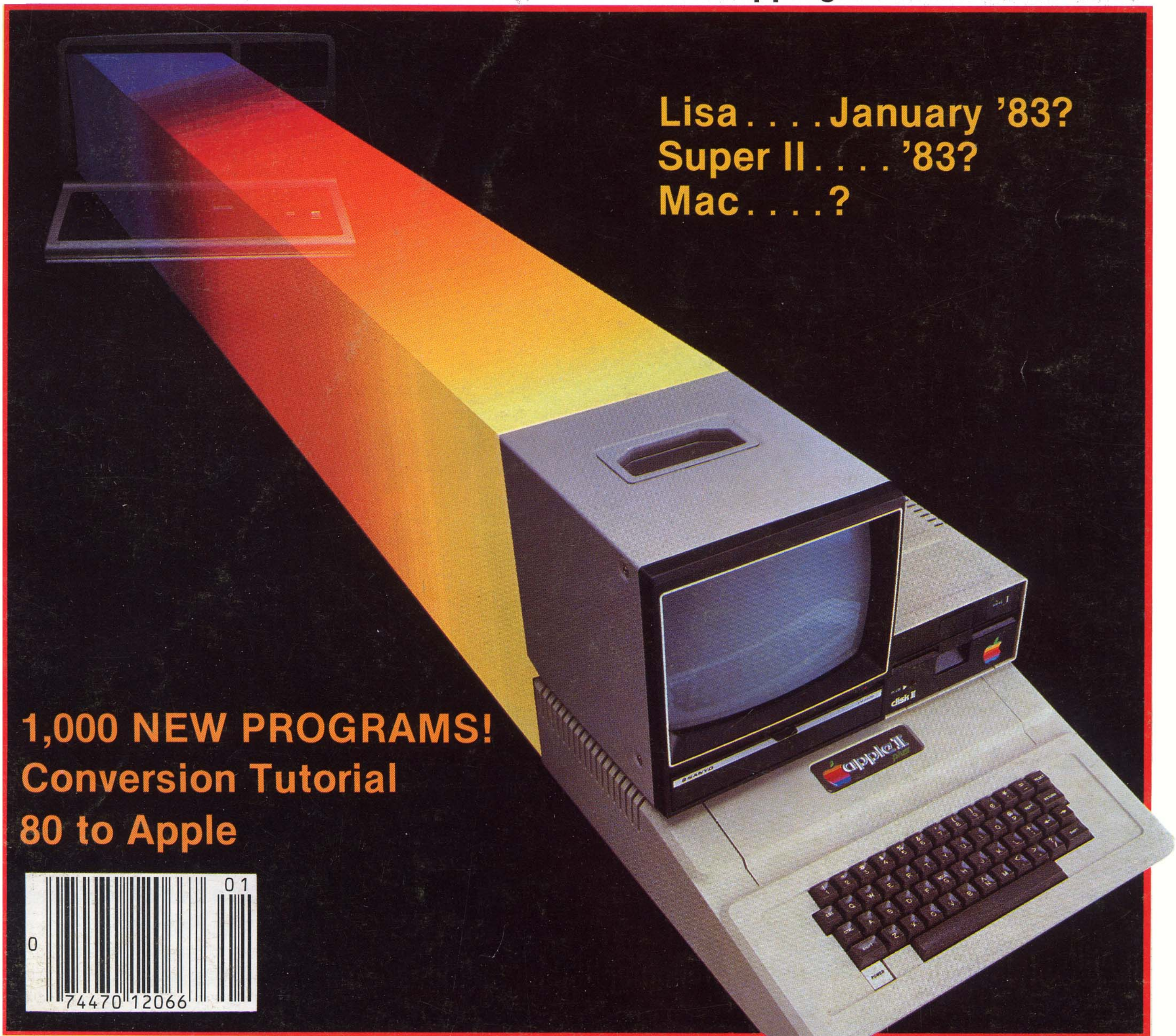
Make Music With Your Apple ///

Data Retrieval Secrets

Peeks & Pokes from Apple's Brass

Trapping Your Pascal Errors

Lisa. . . . January '83?
Super II. . . . '83?
Mac. . . . ?



1,000 NEW PROGRAMS!

Conversion Tutorial 80 to Apple

0 74470 12066 01



SPECIALLY TARGETED. Works in harmony with your z-card/cpm equipped Apple II. No more floppy shuffle. Fast one pass compiler that doesn't generate unnecessary intermediate files which require further assembly.

PROFESSIONALLY DESIGNED. Crafted by a project team with over 15 years of Pascal programming experience. A powerful Pascal that includes all the features necessary for professional program development including random files and dynamic strings.

INCREDIBLE SPEED. Resultant stand alone programs execute 10-20 times faster than the equivalent Basic or up to 4-1/2 times faster than U.C.S.D. programs.

	TRADEMARK
CPM	Digital Research
Apple	Apple Computers
UCSD	University Calif. San Diego
Alcor Pascal	Alcor Systems

POWERFUL TEXT EDITOR. Has Macro capability, reconfigurable and understands smart terminals.

EXCEPTIONAL DOCUMENTATION. Includes 250 page documentation pkg. that guides you step by step.

OPTIONAL ADVANCED DEVELOPMENT PKG. Includes Optimizer and Native Code Generator for even faster executing programs. ADP: $125 + *shipping*.

ALCOR PASCAL: A new standard of excellence from the professionals at Alcor Systems.

CALL OR WRITE for a free 20 page technical brochure with all the facts. 214-226-4476. Pascal System: $199 + shipping.

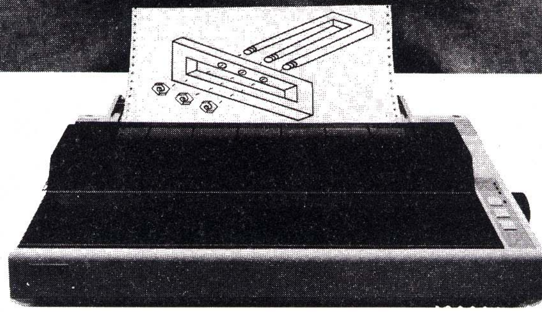
ALCOR SYSTEMS

Alcor Systems
800 W. Garland Ave.
Garland, Texas 75040

Dealer Inquiries Invited

Circle 1 on Reader Service card.





ACTUAL APPLE II PRINTOUT USING GRAPPLER AND EPSON MX100.

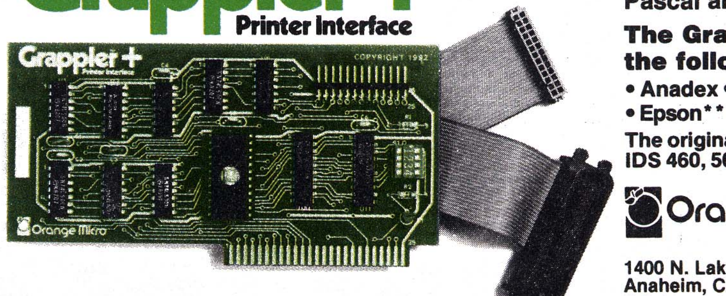
Completely Redesigned. Now, the Grappler+.
The original Grappler was the first graphics interface to give you hi-res screen dumps from your keyboard. The new Grappler+ with *Dual Hi-Res Graphics* adds flexibility with a side-by-side printout of page 1 and page 2 graphics.

Interfacing the Grappler+ to a wide range of printers is easy as changing a dip switch. 4K of exclusive firmware makes the Grappler+ the most intelligent, full-featured Apple® Printer Interface made. And, the Grappler+ is Apple III compatible.*

Up to 64K Buffer Option
An optional Bufferboard can now be added to all existing Grappler and Grappler+ interfaces. See your Apple Dealer for details.

*Requires additional software driver.
**Requires graphics upgrade.

With The Grappler™+ Printer Interface



CPM is a registered trademark of Digital Research, Inc.
Apple is a registered trademark of Apple Computer, Inc.

The Grappler+ Features:
• Dual Hi-Res Graphics • Printer Selector Dip Switch • Apple III Compatible* • Graphics Screen Dump • Inverse Graphics • Emphasized Graphics • Double Size Picture • 90° Rotation • Center Graphics • Chart Recorder Mode • Block Graphics • Bell Control • Skip-over-perf • Left and Right Margins • Variable Line Length • Text Screen Dumps.

The Grappler+ also works with Pascal and CPM.

The Grappler+ Interfaces with the following printers:
• Anadex • Centronics • Datasouth • Epson** • NEC • C.Itoh • Okidata**

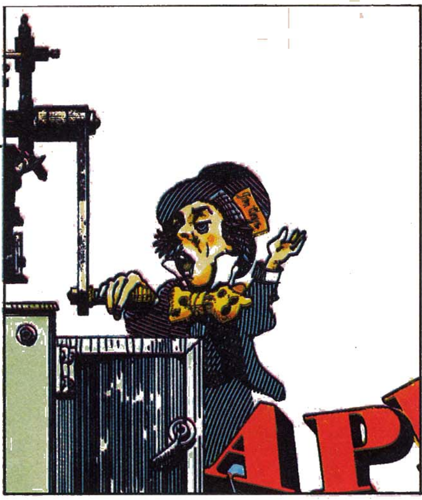
The original Grappler is available for IDS 460, 560, Prism, Microprism.

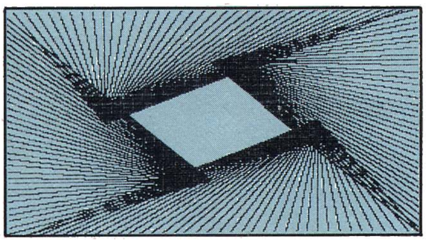
Orange Micro Inc.

1400 N. Lakeview
Anaheim, CA 92807
U.S.A.

Circle 5 on Reader Service card.

inCider
January 1983
Vol. 1 No. 1


Apple/80, p. 96


Bitz, p. 160

ARTICLES

DEPARTMENTS

Cover photography by Martin Paul Studios

inCider (ISSN pending) is published 12 times a year by 1001001 Inc., 80 Pine St., Peterborough, NH 03458. Phone: 603-924-9471. Second class postage paid at Peterborough, NH, and additional mailing offices. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada and Mexico, $27.97—one year only, U.S. funds drawn on a U.S. bank. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. BC Canadian distributor: Graymar Data Services, Ltd., #4 258 E. 1st Ave., Vancouver, BC V5T 1A6. Foreign subscriptions (surface mail), $44.97—one year only, U.S. funds drawn on a U.S. bank. Foreign subscriptions (air mail), please inquire. In South Africa contact *inCider*, P.O. Box 782815, Sandton, South Africa 2146. All U.S. and Canadian subscription correspondence should be addressed to *inCider*, Subscription Department, P.O. Box 911, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form-3579 to *inCider*, Subscription Services, P.O. Box 911, Farmingdale, NY 11737.

WE'VE GOT 24 HOURS TO GET OUT OF TOWN!



Under the Gun?
Needed it yesterday? It's the same old story. However at **QUICKSOFT** we don't mind. We are the fastest, lowest priced software house in the Northern Hemisphere. **QUICKSOFT** is the software house with the difference, **SPEED!** When you place a phone order with **QUICKSOFT** we Guarantee that it will be shipped from stock within **24 hours** or you will receive an additional **5% OFF** the regular **QUICKSOFT** Price.

FREE DELIVERY!
QUICKSOFT offers free delivery on all orders of $50.00 or more.* We not only ship within **24 hours,** but we pay the freight. Plus NO SURCHARGE on VISA and MASTERCARD orders. For placing an order or requesting a **FREE CATALOG** simply dial the **TOLL FREE** number below! **QUICKSOFT** has them **ALL,** and right now we're having specials on these items from our **MOST WANTED** list. *Foreign and Airmail will be extra. C.O.D. orders will be accepted with an additional charge of $5.00 per order.

★★★★★★★★★★ MOST WANTED LIST ★★★★★★★★★★

	LIST PRICE	QUICKSOFT PRICE
PIG PEN	$29.95	**$23.95**
APPLE MECHANIC	29.50	**23.95**
LAZER MAZE	29.95	**23.95**
COMPLETE GRAPHICS	69.65	**55.95**
STAR MAZE	34.95	**27.95**
SNOOPER TROOPS I	44.95	**35.95**
SPELLING BEE	39.95	**31.95**
GERTRUDE'S SECRETS	75.00	**59.95**
MATHEMAGIC	89.95	**71.95**
LETTER PERFECT	149.95	**119.95**
STARCROSS	39.95	**31.95**
TYPE ATTACK	39.95	**31.95**
PFS: GRAPH	125.00	**99.95**
SUPERTEXT 40/80	175.00	**139.95**
ZERO GRAVITY PINBALL	29.95	**23.95**
CHOPLIFTER	34.95	**27.95**
CROSSWORD MAGIC	49.95	**39.95**
GRAFORTH II	75.00	**59.95**
HIGH ORBIT	29.95	**23.95**
NIGHTMARE GALLERY	34.95	**27.95**
ELEPHANT DISKS (BOX)	25.00	**19.95**

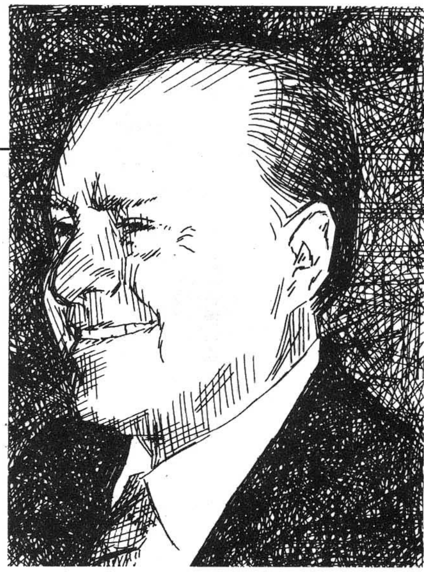
Prices valid thru this issue.

1-800-547-8009 IN OREGON CALL 1-342-1298

QUICKSOFT™

Dept. 31, P.O. Box 10854, Eugene, OR 97440

Circle 27 on Reader Service card.



Remarks from the Publisher...
Wayne Green

In the Beginning

There it was...the first Apple, sitting on a small workbench in the garage of the Jobs home in Cupertino. I'd been making the rounds to see if there was any enthusiasm among the microcomputer manufacturers for me to start a new magazine for the field. A small ad for the Apple got my attention, so I looked up Steve Jobs to see what he had.

Steve, who was living with his parents, called Steve Wozniak to come over so they could show me the computer. They looked like your average young computer hobbyists, complete with jeans, hastily donned T-shirts, and bare feet. But their computer was far simpler in design than those already on the market—it was all built on a single board instead of being spread out over a half dozen plug-in boards. They built it that way because they didn't have the money to go the usual multi-board route, and they ended up with a new type of microcomputer which would be cheaper to make and easier to test than those on the market.

That was in early August 1976, just a few days before the now-famous first microcomputer show in Atlantic City. I recommended that they take a bus, if they had to, and bring the Apple to the show. It looked to me as if this new design could be a winner.

When the show started, there was Steve Jobs, right across the aisle from my booth, showing his Apple prototype. Having been encouraged by the industry leaders to start a new magazine, I was selling prepublication subscriptions to *Microcomputing*. There's a long, interesting story in that, too, but this column is about the Apple, not my publishing history.

At the end of the show Steve was delighted to tell me that he had gotten 20 orders from dealers for his computer and that he was definitely in business. Having pushed so hard to get him there, I was relieved to hear of his success...and 20 computers back in 1976 was a big deal.

A few months later, on another trip to California, I stopped by to see how the Steves were doing. By now they were in an industrial complex and were hard at work on the Apple II. The prototype of the plastic cabinet, shaped like a portable typewriter, was innovative. The color graphics were a real grabber. This was just shortly before they connected with Mike Markkula and the Apple began to take off for the stars.

A year later we began to see more single-board systems such as the Commodore PET and the TRS-80. Just about all of the existing computer stores were selling the Apple—there just weren't very many stores. The first one opened in August 1975 in Santa Monica, pushing the first microcomputer system to be marketed in quantity, the Mits Altair 8800. By August 1976, when the Apple was born, there still were only 50 computer stores in the country. A year later the count was about 500, and then all hell broke loose when Radio Shack started loading their 8000 stores with TRS-80 computers, backing them up with plenty of television ads to bring the customers in for the 1977 Christmas season.

Commodore, which was the first single board system to be sold complete with a monitor, looked like it might take over the market. Enthusiastic dealers signed up to sell it, and PET clubs were formed all around the country. But with poor deliveries, poor quality control, poor documentation, and massive neglect of the clubs, Commodore squandered their lead. It further hurt dealers when prospective customers were able to get immediate delivery of PETs from a mail order house in Illinois, but faced long waits via PET dealers.

Software firms wanting to sell PET-compatible programs found that rarely did any two PET cassette recorders read the same program tapes, so they converted to the TRS-80 in frustration.

Radio Shack held an enormous lead in the field by virtue of their thousands of stores, but they, too, managed to blow the lead eventually. Apple plugged along, building up sales, cooperating with support firms in the development of accessories and software, getting distribution through more and more of the rapidly expanding system of computer stores. The Shack, meanwhile, seemed to go to lengths to discourage friendly support of their system. Eventually this caught up with them and Apple passed Radio Shack in sales. Not bad for a small computer firm up against a $2 billion merchandising giant!

There were a lot of small computer firms in the early days of the microcomputer, way back in 1976 and 1977. Ninety percent of them are gone now, mostly killed by inexperienced advertising and marketing management. Apple not only survived, but prospered.

In 1979, I visited Japan to see what was happening there in the microcomputer field. Apple computers were everywhere—hand-truck loads of them were being wheeled around the streets of Akihabara, the elec-

tronics section of Tokyo. The Apple was the best selling microcomputer in Japan.

By 1980 Apple sales had slipped in Japan as a result of the NEC color computer, which had better graphics. NEC has failed to figure out about the need for software, so the system has made no significant impact in the U.S.

Perhaps pushed too fast in their announcement of the Apple III system by the timing of the public offering of their stock, Apple made some unwise decisions and the III never really recovered from the disaster. This catalyzed some needed management housekeeping and the company seems stronger. Apple may have emerged overcautious, judging from the slow unveiling of rumored new systems.

The IBM Personal Computer, with its 8-bit and 16-bit architecture, changed the whole industry as far as business oriented desktop computers were concerned. New systems had to be mainly 16-bit oriented yet able to run 8-bit software, or at least comfortable with both modes of operation. Thus, rather than redo the Apple III as they had planned, the research department undoubtedly had to redesign it to build in the 16-bit capability.

It may already be too late for Apple to come out with a rumored low-end system designed to compete with the Atari 400 and T.I. 99/4A computers. These, along with the VIC-20 and the Timex-1000, have dropped in price almost beyond belief, all selling for under $200 in a growing number of toy and discount chains. To sell a new system at that price means taking a loss to build up economies of scale, or else forgetting it. You have to make a lot of computers to get the manufacturing costs down to where a system can be retailed for $179 and have everyone make money all along the line.

Even firms as large as Apple and Radio Shack have found that they are too small to be all things to all people. With the low-end market developing quite separately from the middle market, both firms may have to give up the battle with T.I., Commodore and Atari.

The bulk of the Apple sales are from their still very popular Apple II+ system. This will probably be updated soon, unless events overtake my words and the update is released before my editorial. The Apple II presents a solid value by virtue of the enormous amount of software available for it. While Radio Shack has been aggressively shooting itself in the foot by making each new model incompatible with their previous systems, thus requiring either modifications or even total rewrites of software, Ap-

PUBLISHER/EDITOR
Wayne Green
EXECUTIVE VICE PRESIDENT
Sherry Smythe
GENERAL MANAGER
Debra Wetherbee
ASSISTANT TO PRESIDENT
Matt Smith
ASSISTANT PUBLISHER
Jeff DeTray
CORPORATE CONTROLLER
Roger Murphy
CIRCULATION MANAGER
603-924-9471
Patricia Ferrante
BULK & NEWSSTAND SALES MANAGER
Ginnie Boudrieau

ADVERTISING, 603-924-7138
David Schissler, Manager;
Denis Duffy, Craig Everett: Sales;
Patty Lesser: Advertising Coordinator.
New England Advertising Representative:
John A. Garland, Garland Associates Inc.,
Box 314 SHS,
Duxbury, MA 02332 617-934-6464

PRODUCTION
Nancy Salmon, Manager; Michael Murphy, Assistant. Frances Benton, Phil Geraci, Donna Hartwell, Kim Nadeau, Scott Philbrick, Dianne Ritson, Deborah Stone, Theresa Verville, Karen Wozmak; Ad Coordinators: David Wozmak, Mary Seaver; Advertising Production: Bruce Hedin, Fiona Davies, Jane Preston
PHOTOGRAPHY
Sandra Dukette, Laurie Jennison, Irene Vail, Thomas Villeneuve, Robert M. Villeneuve
TYPESETTING
Sara Bedell, Manager; Marie Barker, Melody Bedell, Michele DesRochers, Jennifer Fay, Lynn Haines, Linda Locke, Nancy Newman, Debbie Nutting, Karen Stewart, Susan Weller
DESIGN
Denzel Dyer, Howard Happ, Susan Donohoe, Laurie MacMillan, Dion Owens, Joyce Pillarella, Susan Stevens, Donna Wohlfarth

ple has made their system changes upward compatible for software.

Radio Shack, to thwart outside firms trying to sell add-ons for their system, set policies that discouraged the kind of support for the TRS-80 that Apple has built up. When Radio Shack put on computer shows, they went to lengths to make sure that no outside firms participated. When Apple organized shows, they welcomed outside suppliers and to some degree subsidized them. Not only has this resulted in far more support for the Apple, but the feeling toward Apple is entirely different from the emotions stirred by Radio Shack.

Even the most avid TRS-80 owners seem to despise Radio Shack. I don't, myself. I realize that they are in the grip of calcified company policies. These were probably valid for their dealings with the predominantly lower income buyers of their electronic toys, but seem to be less than productive when applied to their computer business.

With the elimination of discount mail order sales of the Apple, it should be possible for the firm to rapidly add stores to their distribution network. Discounting of Apples made it almost impossible for any store to get an honest profit from selling them, so Apple found that they were gradually losing their marketing system. Now, with the prices firmed up, dealers can again afford to sell the Apple. Further, by virtue of the quantity of support accessories and software, dealers can make more selling Apple systems than anything else on the market—by a wide margin.

Now. . . *inCider*!

In 1975, soon after the first popular microcomputer was marketed, I noticed a need for a microcomputer magazine. By the first of August that year I had the first issue out and was taking it around personally to the manufacturers in the field. Yes, I started *Byte* and was the first publisher. Long story.

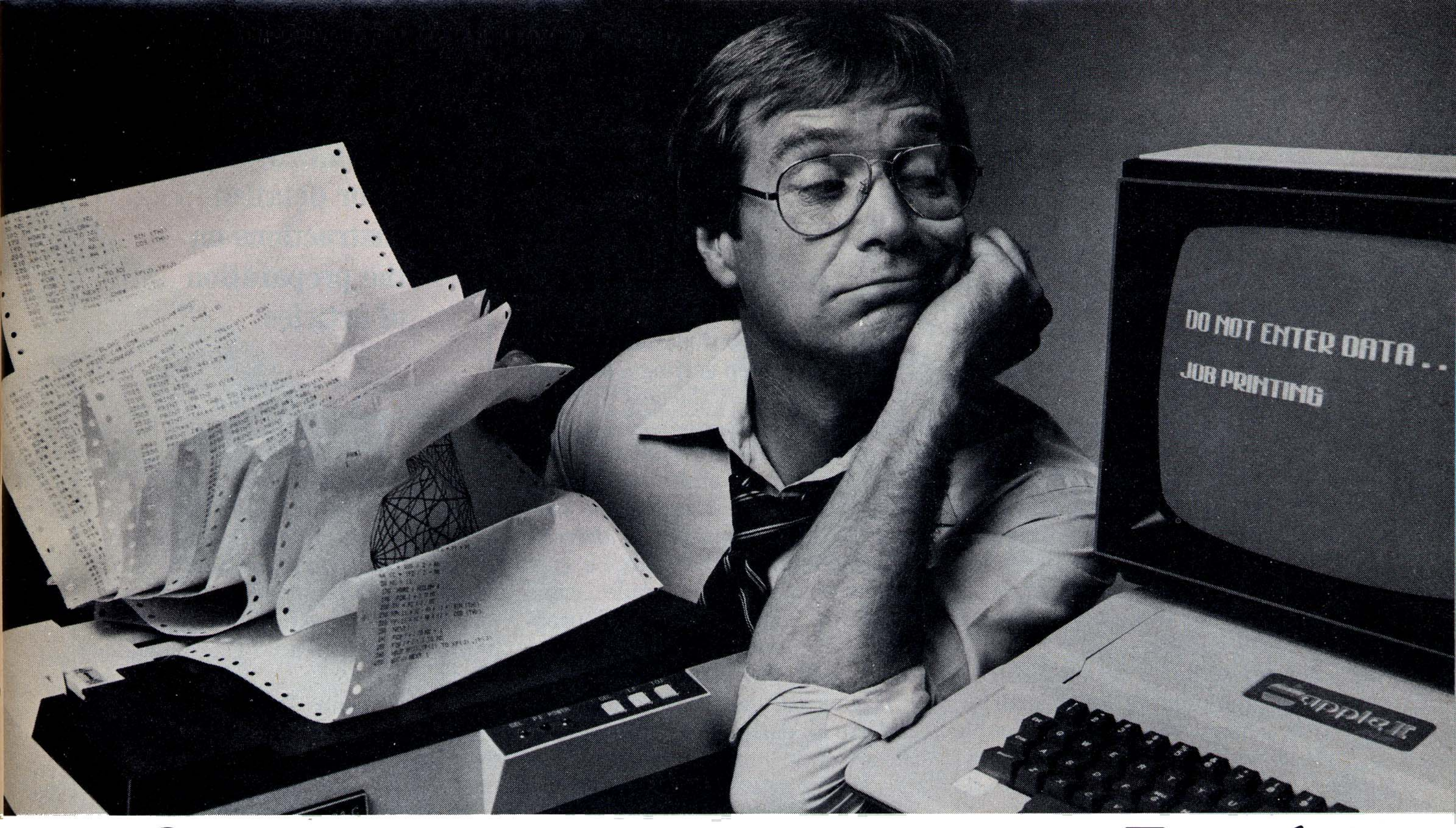
When I lost control of the magazine a few months later through a stock manipulation, I coped with the situation temporarily by adding an I/O section to my ham magazine, *73*. Encouraged during my August 1976 trip, when I first met the Steves, I announced *Microcomputing* later that month at Atlantic City and brought out the first issue in November. A year later, after pushing hard to get someone to go into the mass produced software business, I started Instant Software. The next year I got an industry magazine going, *Selling Micros*. By 1979 there were so many TRS-80 articles in *Microcomputing* that I spun off *80 Micro*, starting with about 132 pages in January 1980. Within two years this had grown to over 400 pages a month, and now it is over 500 a month, with a special 600-page Aniversary edition over and above the regular magazine.

In 1981, feeling that there was a serious need for a completely nontechnical computer magazine for businessmen, I started *Desktop Computing*.

Each of these publications, you'll note, while obviously profitable, has been aimed at helping the microcomputer industry grow. By working out of a small town in New Hampshire I have been able to keep costs extremely low, and thus help new firms to get started with low advertising rates.

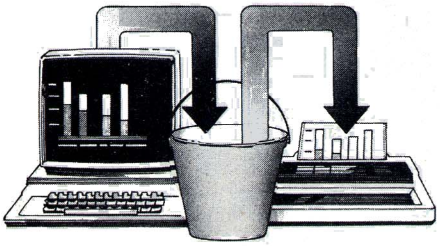
This is in line with the decision to start a magazine to support the Apple. Yes, I know there are several smaller magazines that are Apple oriented, but after waiting for one of them to really stand out, I decided that it was time to provide support of the quality that we have for the TRS-80 computer. So *inCider* will be packed with articles showing you how to get more from your Apple—programs you can run, reviews of software and hardware accessories, and so on. If you've read any of my other magazines you know that the general approach is informal. . . perhaps more like a club newsletter than a pontifical technical journal.

You are encouraged not only to read and enjoy *inCider*, but to write articles or letters on anything you've



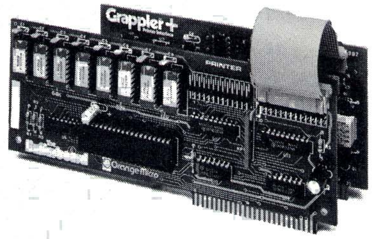
If your printer uses your Apple® more than you do, you need The Bufferboard.™

If your Apple is locked into the "PRINT" mode so much that you've taken up solitaire to kill the boredom, you need a buffer. And if your computer is the Apple II or III, the only buffer for you is The Bufferboard. Expandable to 64K of storage, The Bufferboard stores an instantaneous **bucketful** of print data from your computer. Then it feeds the data to your printer at its own printing rate. Your Apple is set free from driving your printer and is ready for more data from you.



Take your existing interface—and buffer it!

Only The Bufferboard has a simple Interface-Docking System. No bulky boxes or expensive power supplies are needed because The Bufferboard fits right into your Apple—and **docks** onto your existing printer interface. The result is convenient



and economical buffering of most popular printer interfaces, including the Grappler + ™ interface, Epson interface, and Apple printer interface. Thirty seconds and a single hook-up are all you need to end the printer waiting game forever.

Up to 20 letter-size pages stored at a time.

The Bufferboard comes standard with 16K, and is expandable to 32K or 64K of buffering capacity with the addition of memory chips. This "bucket" will hold up to 20 pages of a print job, allowing you freedom to use your Apple.

The Bufferboard—designed exclusively for the Apple Computer.

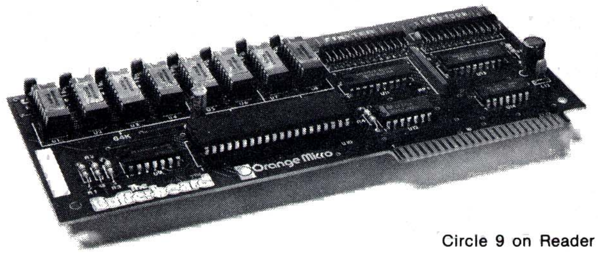
Specifications:
• Versions for Grappler + interface, Epson interface, Apple interface, and other popular printer interfaces • 16K buffer standard • Upgradeable to 32K or 64K • Automatic memory configuration • Automatic self test • Includes interface docking cable.

The Bufferboard is made by Orange Micro, Inc.; the same people who brought you the popular Grappler + printer interface. Both the Grappler + and The Bufferboard are now available at your local Apple dealer.

Apple is a registered trademark of Apple, Inc.
Epson is a registered trademark of Epson America, Inc.

Orange Micro inc.

1400 N. Lakeview, Anaheim, CA 92807
(714) 630-3620, TELEX: TX 183511 CSMA



The Bufferboard™
For Apples and Printers

Circle 9 on Reader Service card.

"You can write for detailed instructions on the preparation of articles..."

found out about your Apple that you think others would appreciate knowing about. Pass along some of your programs to help others—your ideas. Perhaps you've found an interesting application for an Apple. Maybe you've built a gadget that interfaces an Apple with something else. We'd sure like to have a lot more information on using Apples for typesetting, for controlling machinery, writing music, networking—even networking a group of Timex computers using an Apple as a host...it's endless.

You can write for detailed instructions on the preparation of articles, but in general we like them double spaced with generous margins, the best pictures you can manage, and so on. Length? Whatever it takes to tell the story. Don't be too brief and don't lard it out, expecting to be paid by the word. Oh, we *do* pay for articles...pay rather well, matter of fact. That's the icing on the cake. You'll get your fun from helping others, seeing your name in print, and perhaps even getting a substantially higher salary the next time you change jobs.

If you are interested in having fun with your Apple and making some money too, you might look over some of the programs in *80 Micro* or *Microcomputing* that can be converted to run on the Apple...and make the conversions. We'll probably be interested in publishing 'em...with payments.

If you are the adventurous type and are an early buyer of some new accessory, printer, modem or whatever, you might want to review the equipment for possible publication. The payment would help defray the cost. But don't try to pull any fast ones with manufacturers, asking for discounts or freebies on the strength of a published review. We don't go with that.

So, with your help and *inCider*, maybe we can keep Apple right out in front, no matter what Radio Shack or the low end computer firms do. Let's get cracking.

I Need Your Help

Yes, *inCider* will be on the counters of computer stores, in many book stores around the country, on newsstands, but this is still a slow way to get the word out. If we're going to get *inCider* to every Apple owner, I need your help personally. Please show your copy of the magazine to any friends who have Apples. If you belong to an Apple user's group, let's get as many subscriptions as we can. The more readers you help me get, the more advertising we will have—and the more ads, the larger the magazine. With any serious cooperation from you and your friends, we'll be publishing a magazine bigger than *80 Micro*. Then the problem is to get the time it takes to read a 500-page magazine.

Think of the incredible amount of information you'll get...programming short cuts, faster sorting routines, language tutorials so you can fit the language to the kind of program you want to write...coverage of every compatible accessory in depth so you'll not waste your money on bummers, news of new products to make your Apple do more things...run faster, cooler, and so on.

Further, since I'm kind of a bug on helping people to get rich, you'll be getting all sorts of ideas on ways to escalate your Apple experiences into business opportunities. There are more ways of getting wealthy today than in any time in the past, and microcomputers are a key to grabbing the golden ring. Stick with me and pay attention. If you don't have the enthusiasm for getting rich, perhaps your kids will do it for you. Just look at Steve Jobs, listed recently in the *Forbes Magazine* list of the 400 wealthiest Americans!

One need today, according to the people in my Instant Software division, is for programs for the low end computers. There's gold in those $100 and $200 contraptions if you like to write programs. The main need is for games, but educational programs, simulations and so on sell too. Instant Software has the organization to get your programs into finished shape, publish them, package them and get them into thousands of stores. They'd love to have a couple hundred good programs for the Timex, or any of the others.

One More Thing

When you are submitting articles for *inCider* we do need a printed copy, but, if you are using your Apple for word processing, our publishing process can be speeded up if you also send in the copy on a disk. This will save time, allowing us to edit the material and then feed it directly into our typesetting system. ■


Circle 23 on Reader Service card.
COMPUTER GRAPHICS
AMDEK
apple II
for the Artist
Special Effects
by David Lubar and Mark Pelczarski
Paint on your computer in over 100 colors with 96 different brushes! Magnify mode for precision touch-ups; mirror image; color reverses; move any part of picture anywhere; packing routine for increased disk storage of any standard graphics screen. Can be used by anyone. Paddle/joystick $39.95. Apple Graphics Tablet $69.95.
for the Designer
THE COMPLETE GRAPHICS SYSTEM II
by Mark Pelczarski
Everything needed for computer-aided design. Easily draw and design in two or three dimensions, mix text and graphics freely, define shapes and create typefaces with unique character generator. Great for presentations, videotape displays, storing designs on disks, graphics for other programs, or just for experimentation. Paddle/joystick $69.95, Apple Graphics Tablet $119.95. 50 additional fonts, $19.95.
for the Programmer
The Graphics Magician
by Mark Pelczarski, David Lubar, and Chris Jochumson
Add fast, smooth animation and hundreds of pictures to your programs. Create animated shapes and draw paths; have up to 32 independently controlled figures on screen. Use a special palette of over 100 colors to create pictures and objects in highly compressed format. Easy to use, and includes a special programming tutorial. Has saved months in development time for several games already on the market. Paddle/joystick $59.95, Apple Graphics Tablet $69.95.
penguin software 830 4th Avenue, Geneva, Illinois 60134 (312) 232-1984
All Penguin applications products are now on unprotected disks for your convenience. Apple II is a trademark of Apple Computer, Inc.

inCider

inCider editorial offices
Pine Street
Peterborough, NH 03458
603-924-9471

Problems with Subscriptions: Send a description of the problem and your current and/or most recent address to: *inCider*, Subscription Department, PO Box 911, Farmingdale, NY 11737.

Change of Address: Send an old label or a copy of your old address and new address to: *inCider*, PO Box 911, Farmingdale, NY 11737. Please give eight weeks' advance notice.

Microfilm: This publication is available in microform from University Microfilms International. United States address: 300 North Zeeb Road, Dept. P.R., Ann Arbor, MI 48106. Foreign address: 18 Bedford Row, Dept. P.R., London, WC1R4EJ, England.

Dealers: Contact Ginnie Boudrieau, Bulk Sales Manager, *inCider*, Pine Street, Peterborough, NH 03458. 800-343-0728.

Let's Take a Peek . . .

Welcome to *inCider* magazine. We have plenty to offer you. At the head of the list are articles written by key personnel within Apple Computer Inc. These "incide" views and programs will benefit all of us, by giving us a peek inside Apple, and useful articles and programs to help us understand our system.

Commencing with this issue is an Applesoft tutorial entitled The Applesoft Adviser. This monthly column will offer blocks of information for novice and experienced programmer alike. You won't have to wait for the February issue to complete January's task.

Pascal receives ample space, and starting next month, a Pascal tutorial will aid in your understanding of the language. Assembly, Forth and Logo are all coming!

The *inCider* educational department will report on all efforts to use computers for learning. Need information on what's new for the Apple? Check the New Products and New Software departments, edited by Linda Stephenson, for concise descriptions of the latest offerings. Software and book reviews will offer educated opinions on the latest releases; and hardware reviews will help you determine the best buys in add-on peripherals. News more to your liking? Read John Mello's Apple Watch. He covers the latest information and what makes the industry tick.

Apple III users, *inCider* has a column for you as well, called III's Company, written by Bill O'Brien. Also within are graphics and music programs for your Apple III, the latter written and coded by a talented ten-year-old. A featured business column commences soon, and two articles in this issue should appeal immediately to the business professional. Hints 'n' Techniques will make your Apple even more enjoyable. Graphics departments, such as Bitz and Graphic Goodies, are also regular attractions.

You'll find the stuff and nonsense of Paul Raymer's outCider. There's a hardware modification for Epson owners who need just a little more from their interface. Utilities, as well as a review of the Micro Sci A40 and A70m drives, are bonuses. Names to watch for on a continuing basis are Lee Sumner, Fred Huntington, Gregory Glau, Hap Gaylord, David Kutzler, James Florini, Jerry Brieger and Paul Schubert.

Your name could appear in an *inCider* table of contents. We seek submissions on any topic related to the Apple computer and its allies. Send your manuscript double-spaced, typewritten, with a self-addressed, stamped envelope in case we need to return the submission for changes. If you have a program you'd like us to see, send it on. Include a program disk. We also need photographs and/or schematics of all hardware modifications and constructions.

We're going to drive the stuffies away, so put humor in your writings. Like to draw? We are on the search for top-flight cartoons that play with computing in some general or specific manner.

Arcade and adventure games always have a high billing with *inCider*. All work and no play . . . you know the bit.

Your letters are also highly valued. Factually speaking, the readers will determine the course *inCider* takes in future issues. What you want, you'll get. But you have to tell us what you'd like to see. Better still, write an article.

Send your manuscripts and programs to:

Editorial Department
inCider Magazine
80 Pine St.
Peterborough, NH 03458

Thank you for reading *inCider*. We hope you enjoy it and come back next month for more.

—the editors.



. . . for demanding graphics or text display.

For high resolution (560H X 240V) color graphics, you can't beat the Amdek Color-II Monitor. And if you're looking for economy, the Color-III Monitor with 260H X 300V resolution is a superb buy.

Both monitors feature RGB video input for computer controlled color . . . and Amdek's easy-to-install Digital Video Multiplexor (DVM) board permits interface with the most popular 80 column boards.

Just call, or write for full details.

- Color-II Monitor has RGB input and 560(H) X 240(V) line resolution.
- Color-III Monitor has RGB input and 260(H) X 300(V) line Resolution.
- Digital Video Multiplexor (DVM) assures color graphics interface with most popular 80 column boards, such as: Videx "Videoterm", Advanced Logic "Smart-term", M&R "SUP 'R' TERMINAL", Bit-3 Computer Corporation "FULL VIEW-80", and the "Doublevision" boards.

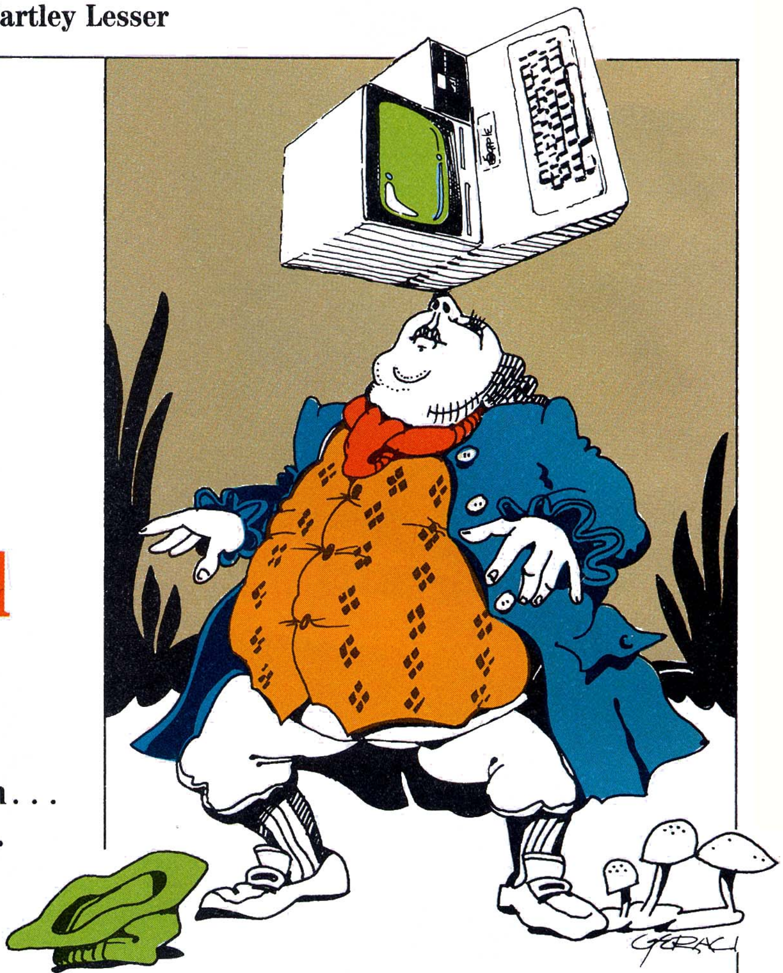
2201 Lively Blvd. • Elk Grove Village, IL 60007
(312) 364-1180 TLX: 25-4786

AMDEK CORP.

Amdek . . . your guide to innovative computing!

Circle 95 on Reader Service card.

by Hartley Lesser



Simply Stated

Words of praise, and words of concern... our Technical Editor rants and raves.

A Green Apple? Why not? At least this one won't upset your stomach, like some others do. What better way to digest Apple and Apple compatible data than through *inCider* magazine. Best of all, *inCider* is entertainment! Though a tall order, *inCider* will be the information clearinghouse for all Apple end-users.

Hate that last word, end-user? Sounds degrading, doesn't it? As if the user is a bottom-of-the-barrel kind of person, a real dreg. End has evolved a somewhat terminal image, but contemporary micro users are intellectually and personally alive and thriving. As with any other business, computing has its buzz words. Authors smatter their articles with jargon and believe they know more than the readers. Like most fallacies, such thinking should be flushed away. So, when you notice a buzz-word in this column, it's just that no other phrase works as well, and the word has standard meaning in the industry.

Let's start zapping some of these overused catch-alls. *Smart-user* is far better than end-user. After all, you've either purchased, or borrowed, the finest system on the market—powerful, flexible, with the broadest and most reliable software and hardware base in the microcomputing field. Best of all, the Apple system is user-friendly! (Ugh, another buzz.)

One reason Apple has earned such a high share of the computer market is that those who own the machine are loyal. Whether this loyalty is budget-motivated or not makes no difference. Of course the wealth of Apple peripheral enhancements and professional/entertainment software also makes a direction change less than appealing. One gets addicted to the best. Another sign of intelligence.

What further proof of your smart-user mentality do you need? At this very second. . . a copy of *inCider* is in your hands! No ordinary computer magazine, this Premiere Issue. Have you thought what such a magazine might be worth three, five, ten years from now? Maybe twenty. . . no, thirty bucks? Look at what those 40s superhero comic books are bringing at comic book conventions. And *inCider* has a definite advantage over those cartoon characters. . . everything you read here is part of computing history. Not many extra copies will be just lying around, and computerphiles will want to own the Apple magazine that adapted itself to its readers' needs. Ten years from now, the utilities and tutorials you enjoy in this collector's copy could well have started you on a career that leads to the very pinnacle of success! Just don't forget us small folk when you bring in your first million, O.K.?

I'm excited about *inCider*. Why?

First, we intend to listen to you, the reader. What you want in the way of Apple articles, *inCider* will obtain for you. Naturally, part of the responsibility is yours. Write the material you want to see published. I guarantee that the reason a new programming language, or a certain word processor, or an optimizer, is missing from *inCider* is that you haven't taken the time to write such an article.

Every manuscript submitted will be read thoroughly. If some submissions show promise, personal letters will accompany their return to the author, suggesting ways to improve the article. Hobbyists have written some of the best material *inCider* has received, as has a neglected core of businesspeople who manage, write and produce the software and hardware you enjoy on your Apple.

Second, *inCider*'s reviews are consumer bulletins. Both our in-house and contributing reviewers have nary a connection with any production company supplying the software. Should a review prove negative, the company will have a chance to rebut the review. After all, sometimes personal favorites color succeeding reviews, thereby diminishing the product's chances at gaining the favor of the author. If you feel you're not seeing enough negative reviews, please keep in mind that we're talking about Apple computer products here. A very unwise businessperson markets a lousy program!

Our columnists are top-notch, and within the next month or two some surprises will appear in *inCider*. The crew of three that managed to paste and pull this Premiere Issue together are now beginning to tread water. We can promise that, through mutual cooperation, *inCider* magazine will set new standards for microcomputing periodicals.

The Grapevine

What of these rumors concerning new Apples? Apple is playing more than adequately close-mouthed. I have no wish to quibble with Apple Computer Inc., but it does seem as though they are totally neglecting one type of system, the low-end system. Apple has no entry in this volatile, profitable and mammoth market.

Granted, when Super II, Lisa, and Macintosh are released, Apple will reap broad benefits, from an extended user base to compounded peripheral and software development and sales. Perhaps they think they have already lost the market to the currently-available low-end systems. Unless they move quickly, that will be true.

Apple and IBM users,

Wizard

performance products add magic to your computer

Wesper Microsystems has created the finest of plug-in performance products for your Apple or IBM Personal Computer. The Wizard products truly add magic to your computer.

WIZARD-80 *(for Apple II)**
Instant Video 80-Column Card
Doubles your vision...Lets you see exactly what you will get when typing/editing 80-column format. Displays upper/lower case. Compatible with CP/M, word processors, modems, software and all current Apple II expansion boards.

WIZARD-16K *(for Apple II)*
16K RAM Memory Card
Expands your memory...Adds 16K bytes of RAM. Compatible with Z-80 cards for CP/M. Runs with all Apple languages.

WIZARD-BPO *(for Apple II & Parallel Printer)*
Buffered Printer Output Card
Sets your Apple free...it lets you continue using your Apple while the printer prints. Contains 16K byte "Character Buffer" expandable to 32K bytes. Graphics...of course.

WIZARD-SOB *(for Apple II & Serial Printer)*
Serial Output Buffer Card
Releases your Apple...You can use your Apple even while the printer prints. Features are same as Wizard-BPO but for use with serial (letter quality, etc.) printers.

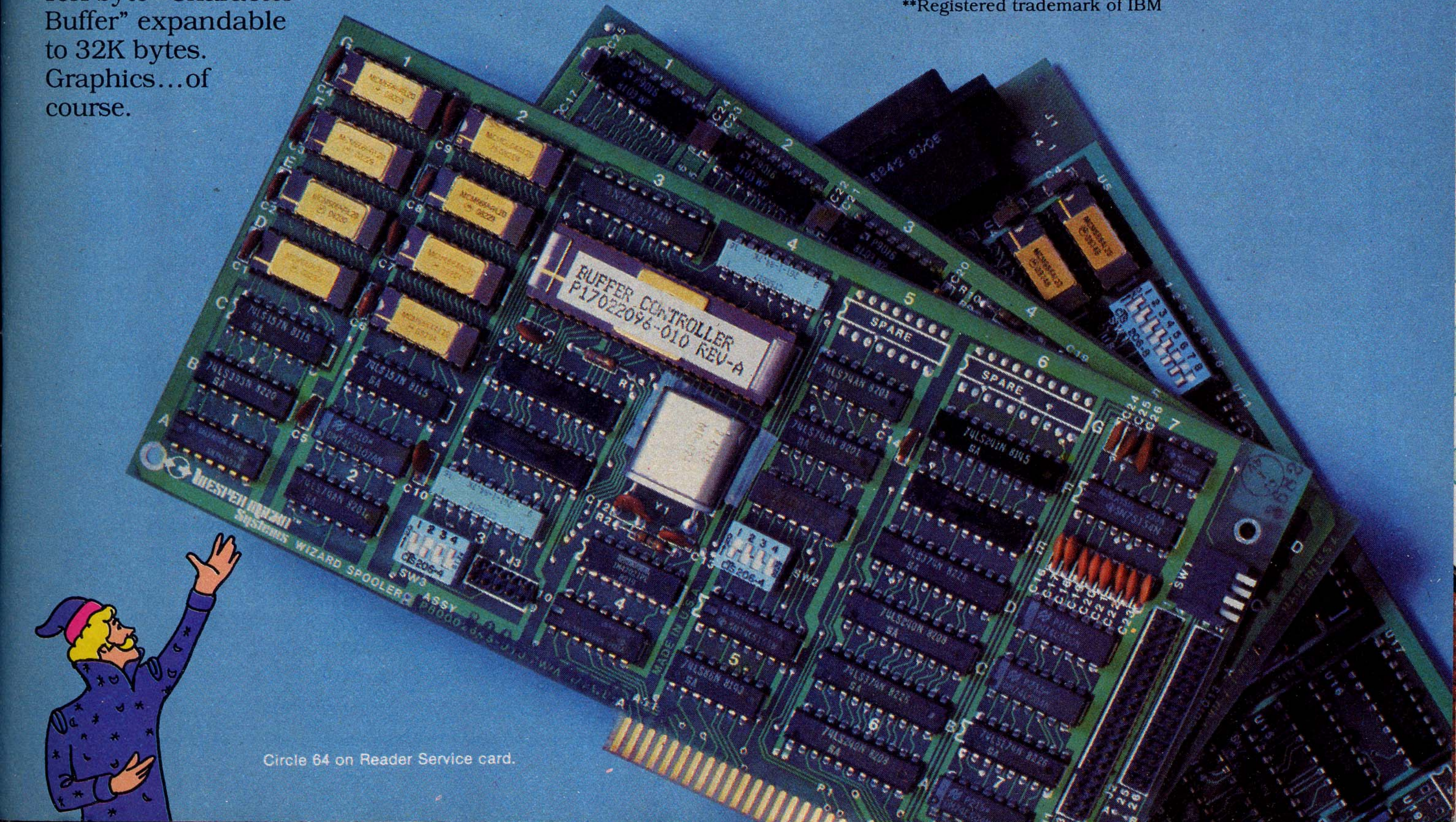
WIZARD-Spooler *(for IBM PC)***
Full Feature Spooler Card
Parallel printer interface. 16K buffer with 32K or 64K optional. Lets you input new task while old task is being printed.

All the Wizard's products are covered by a two-year Factory Warranty.

Available at all fine computer stores.

WESPER MICROSYSTEMS™
A WESPERCORP COMPANY
3188 Pullman Street
Costa Mesa, CA 92626
(800) 854-8737 (714) 850-1666

*Registered trademark of Apple Computer Inc.
**Registered trademark of IBM



Circle 64 on Reader Service card.

Thousands of consumers would just as soon spend the money on a basic computer system as on a video game machine. Granted, the gossip is that some retail dealers are *dropping* one product because their actual profit margin is so slim as to be nonexistent. But Apple has nothing to fill this void. Which means a second computer system will be sitting atop the hill, grinning from ear to ear.

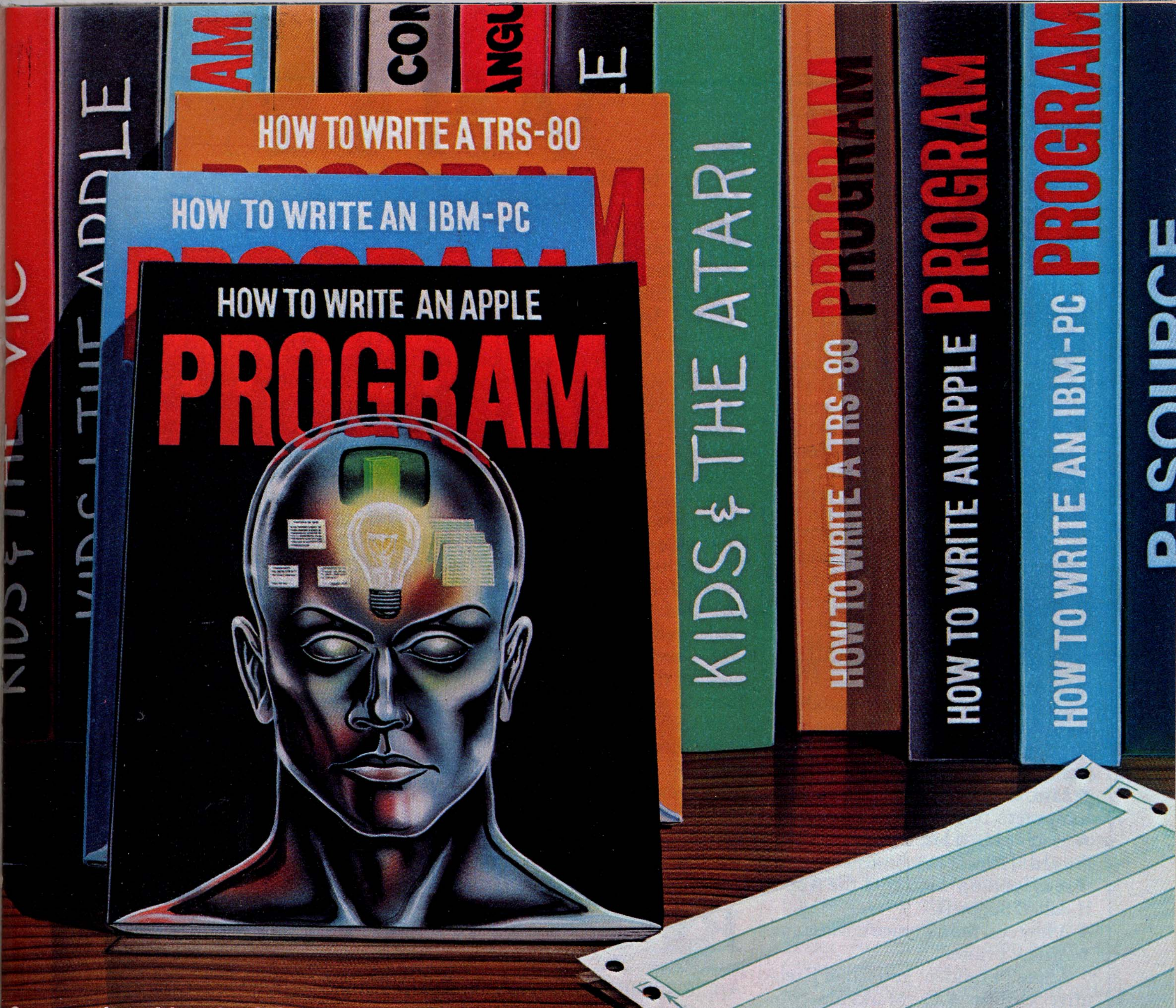
Apple would bring into the low-end market a reputation second to none. Their machine would probably sell for a tad more than the current leader, but with quality software and inexpensive peripherals, Apple would soon have today's favorites looking on enviously at its rising market share. To reiterate, if a low-end system isn't produced soon, they might as well forget it. Not every computer consumer wants two disk drives, multiple language capabilities and 64K bytes of RAM. A machine that could run his or her favorite games, perhaps manage a checking account, or maintain a simple database of library books or records for less than $500, would be an instant success.

''Only the folks inside Apple know what's truly going on. Speculation is the name of the game.''

Only the folks inside Apple know what's truly going on. Speculation is the name of the game. I can guarantee that when *inCider* learns of new developments, either about Apple's three new computers, or a low-end system, you'll be the first to know.

The magazine is not sponsored by Apple Computer Inc., and that corporation holds no editorial or professional sway over this magazine. Remember, this is a magazine for the Apple. Even-handed editorial policy will ensure that *everyone*, including Apple, gets a fair shake.

I hope you enjoy your Premiere Issue of *inCider* magazine. Write and tell us what you think. This magazine will become the reader's outlet, and communication is the name of the game. Special thanks to all who worked extremely hard to produce *inCider*. ■



3 exceptional books join the DATAMOST library.

INTRODUCING . . .
HOW TO WRITE A PROGRAM

Here is a series of easy to read, easy to use, easy to understand books, which teach you how to write usable, useful programs on your computer. And you don't have to worry about irrelevant material which has no interest for you, because there are three specific volumes. One for the Apple,* one for the IBM-PC,* and one for the TRS-80.*

In each of these books author Ed Faulk leads you through your favorite computer and takes the mystery out of writing programs for it. As you proceed, interesting chapter by interesting chapter, you'll wonder why you were ever intimidated by the thought of programming!

If you want to get the very most out of your Apple, IBM-PC or TRS-80 then you really want HOW TO WRITE A PROGRAM. Before you're past Chapter 2 you'll be programming. By the end of the book you'll be willing to tackle business programs, personal use programs and even games and adventures! **$14.95**

Get your copy now. Available at computer and book stores, or:

DATAMOST INC. (213) 709-1202
9748 Cozycroft Ave., Chatsworth, CA 91311

*Apple is a trademark of Apple Computer, Inc. IBM-PC is a trademark of IBM Corp. TRS-80 is a trademark of Tandy Corp.
VISA MASTERCHARGE accepted. $2.00 shipping/handling charge. (California residents add 6½% sales tax.)

Circle 75 on Reader Service card.

Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia
Toll free (800) 336-0338

edited by John P. Mello, Jr.

New Apples in '83 Basket

Lisa, Macintosh, Super II hoped to heal soft spots in Apple domain.

More than three new products will be introduced this year by Apple Computer Inc. They include:

- Lisa, a high-end personal business computer;
- Macintosh, a less expensive, more limited business machine;
- A souped-up version of the Apple II called Super II;
- A low cost mass storage device; and
- A new printer.

Apple's President and Chief Executive Officer A.C. Markkula confirmed to *The Wall Street Journal* that the debut of Lisa would be held this month.

The machine is expected to be very easy to use. A first-time user can be computing on it in 20 minutes, according to a report in the *Boston Globe*.

New products are the key to Apple's ability to compete in the desktop computer market, one analyst told *Computerworld*. Yankee Group President Howard Anderson predicted IBM would dominate that market by 1986. Apple will lose its front spot, he said, by failing to introduce new products.

Another analyst, Esther Dyson, vice president of Rosen Research Inc., told the *Journal* the new products could massage strained relations between Apple and its independent dealers.

"If the new products are good," Miss Dyson said, "it will take care of a lot of dealer unhappiness."

According to the *Journal*, tensions between Apple and the dealers stem from the Cupertino, CA, firm's installation of an in-house national-accounts program to handle major sales to Fortune 1000 companies.

Dealers also burned at Apple's inability to control sales of its computers by unauthorized dealers. By underselling authorized dealers, the mavericks force those dealers to cut their prices and make less money on the machines.

The *Journal* said the mavericks buy their Apples from authorized dealers who overbuy. Those dealers add a few dollars to the price they paid for the micros, then secretly sell them to an unauthorized seller.

"Apple is firmly committed to strengthening the integrity of its authorized dealer network," CEO Markkula told the business daily. He would not elaborate on that statement, the *Journal* said, but industry watchers believe Apple is preparing strong measures to choke off supplies to the outsiders.

Overall, Markkula said, Apple is "in very, very sound financial shape."

He estimated a 1982 fourth quarter surge of 70 percent in profits and 80 percent in sales, a surge sparked by aggressive promotion of its personal computer products.

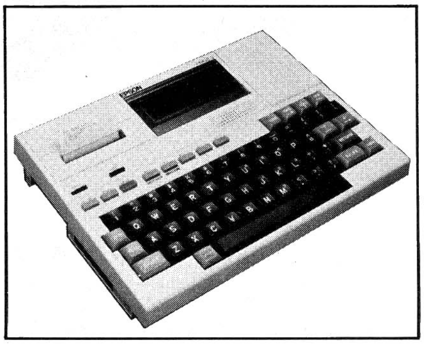
Those gains indicate fourth quarter earnings of some $19 million, or 32 cents a share, on sales of $176 million. That would be up $11 million, or 19 cents a share, on sales of $97.7 in fiscal 1981.

For all of fiscal 1982, Apple's earnings climbed 50 percent with a 75 percent boost in sales, Markkula said. It is estimated the gains would hike profits to more than $59 million, or about $1.05 a share, from $39.4 million, or 70 cents a share, in fiscal 1981. Sales would have risen to almost $586 million from $334 million in fiscal 1981.

"I believe we have a chance to break into the Fortune 500," Markkula told the *Journal*. He added he believed that would be a record for a company that's been in existence for only five years. ■

Execs Bolt EuroApple

Apple's complacency sends Old World executives packing



Epson HX-20: Attracted Apple executive.

Rifts between Apple's Cupertino, CA, headquarters and its European outposts have been blamed for a rash of defections from the firm's offices on the Old World side of the Atlantic.

According to a UK fortnightly called *MicroScope*, Apple's complacent approach to the European market has caused several of its key people to bolt from the firm, including its head honcho in Europe, Tom Lawrence.

Lawrence resigned from Apple to join a Sunnyvale, CA, firm, Valid Logic Systems. *MicroScope* reported Lawrence was unsatisfied with Apple's policy of tailoring its computers for European markets.

Apple's managing director of European manufacturing, Alex Wrafter, also bolted the company. The quoted reason for his departure: disagreements with company policy.

In Apple's UK operation, sales manager John Patterson left the firm to set up Epson's HX-20 micro in the British market.

"To lose one senior manager is a misfortune," the fortnightly said. "To lose four looks like carelessness." ■

Circle 46 on Reader Service card.

A TEACHER FOR THE APPLE

Learn *How to Operate the Apple II Plus* with the same self-study course Apple has chosen to train its own field sales reps!

Imagine having a personal tutor "talk you through" the practical operations of an Apple II Plus computer. In just 3 "hands on" training sessions of one or two hours each, you'll learn how to:

- Use each special key and essential command
- Load and run programs
- Save programs and data
- Copy programs and diskettes
- Protect programs
- Modify programs
- Use your computer as a super calculator
- And much more!

No Programming Required. You don't need to be a programmer to operate a computer, any more than you need to know how to design a car to drive one.

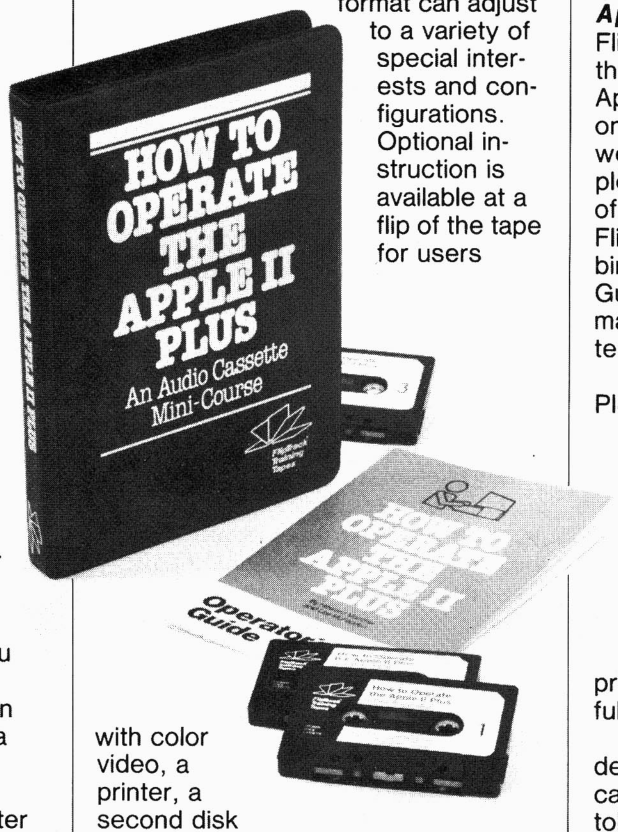
Our purpose is to put you in the driver's seat of your Apple computer so that you can confidently use existing programs . . . enter your own data . . . organize your files . . . and protect them. If you later decide to take up programming, you'll have the necessary background for doing so.

Teach Yourself. Train Others. *How to Operate the Apple II Plus* is designed for the first-time computer user. No technical knowledge is assumed. Perfect for training others: Staff . . . colleagues . . . clients . . . customers . . . students . . . family. Use it again and again.

Why Audio Instruction? Simple. It's human. It's interactive. It's multi-sensory. It's patient. It's inexpensive. And it works!

FlipTrack® Training Tapes deliver carefully sequenced instruction in a relaxed, conversational manner. You concentrate on the screen and keyboard, not a book. You learn at your own pace, trying every command and observing the results.

The FlipTrack® Difference. Best of all, our patented FlipTrack® cassette format can adjust to a variety of special interests and configurations. Optional instruction is available at a flip of the tape for users with color video, a printer, a second disk drive, or game controllers.



You save time and avoid confusion by learning just the procedures that relate to your system, or listening to just the topics that interest you. With FlipTrack® Training Tapes, you choose whether to follow a "full" track or a "fast" track every step of the way.

Examine *How to Operate the Apple II Plus* FREE for 15 days. Let FlipTrack® Training Tapes speak for themselves by dropping by your local Apple dealer for an on-the-spot demonstration. Or if that's not convenient, we'll be happy to send you the complete mini-course on a 15-day "right of return" basis. You'll receive all 3 FlipTrack® cassettes in a handy vinyl binder, along with an Operator's Guide, which will serve as your permanent reference to the procedures, terms and concepts of the course.

Your only prerequisite is an Apple II Plus computer with the System Master 3.3 diskette and a standard audio cassette player. No computer hookup is necessary for the cassette player.

Try a lesson or two yourself, or share them with others. If you are not delighted at how quickly and easily you master the operation of your Apple, simply return the program in its original condition for a full refund. No questions asked.

To order . . . See your local Apple dealer. Return the coupon below. Or call us toll free. Here's our 24-hour toll free number for credit card orders only:

800/227-1617, Ext. 439

In California, call 800/772-3545, Ext. 439.

You have nothing to lose, so act today.

YES! Please rush me *How to Operate the Apple II Plus.* I understand that if I'm not delighted, I may return the mini-course undamaged within 15 days for a full refund. No questions asked.

☐ My check is enclosed for $52.45 ($49.95 plus $2.50 shipping & handling) Illinois customers include sales tax.

☐ Charge my credit card:
☐ VISA ☐ MasterCard Interbank No.________
Card No.________________ Exp. Date________

Name________________________

Address________________________

City______________ State_____ Zip______

Signature______________ Phone__________

Outside USA add $10 per unit. Airmail extra.

Apple is a TM of Apple Computer, Inc. VisiCalc is a TM of VisiCorp.

FlipTrack® Learning Systems

Division of Mosaic Media, Inc.
Dept. C-1, P.O. Box 711
Glen Ellyn, IL 60137
312/790-1117

NEW COURSES!

How to Operate the Apple ///
4 audio cassettes, 1 diskette & Operator's Guide . . . $95.00

How to Use VisiCalc
4 audio cassettes & User's Guide $65.00

Dutch Treat for Apples

Hobbyist group in Netherlands creates package to make Basic programs interchangeable.

Consumers new to the field of desktop computing often express dismay upon learning that programs recorded on an audio cassette by an Apple can't be loaded into a TRS-80, or a Sinclair ZX-81. They are perplexed at the subtle, machine-specific dialect differences in the Basic programming language. These differences may even turn some people away from personal computing.

A group of Europeans has decided to do something about this situation. They have created a standard version of Basic called BASICODE; and, they have created an audio cassette read/write protocol that will work with ten different brands of personal computer.

There are reportedly over 12,000 computer hobbyists using BASICODE in the Netherlands. Developed by a Dutch nonprofit organization called the Hobby Computer Club in conjunction with the producers of a Dutch radio program called HobbyScoop, BASICODE consists of translator software and, for certain machines, a hardware modification called a "converter." The translator software works with the hardware converter to produce the same audio tones for reading and writing cassette information.

BASICODE uses a universal instruction set of reserved Basic keywords. It is essentially a combined superset of existing versions of Basic.

BASICODE is available for the Philips 2000, Acorn ATOM, DAI, Apple II, Exidy Sorcerer, PET/CBM, SWTP, TRS-80, Cosmicos 1802 and OSI C1P microcomputers. The program is usable in machines with from 4K to 48K of memory.

To purchase the BASIC cassette tape and manual, send an international money order equivalent to 30 Dutch guilders ($11.36) to:

BASICODE
Administratie Algemeen
Secretariaat, NOS
PO Box 10, 1200 JB Hilversum
The Netherlands

TRS-80 owners will have to install a fairly extensive hardware circuit board add-on to use BASICODE. The manual gives instructions on how to build the modification, and lists an address where a fully assembled board can be ordered. TRS-80 machine-language programmers may be dismayed to learn that the commands in the source code for the translator program are written in Dutch. ■

—G. Michael Vose

Fast Lane Pascal

Apple Pascal 1.1 can run 94 percent faster with a new package from Legend Industries Ltd. of Pontiac, MI.

The package includes a 128K KDE disk emulator random access memory board, a Stellation Two 6809 Mill, and a Super I disk to contain the software.

The 128K disk emulator card uses only one Apple slot and looks to the computer like an almost instant access drive. The 6809 card plugs into an Apple slot. The two cards are tied together by the Super I software on disk.

The price of the speed-up package is $749. ■

Apple Bites Back

Makers of Pineapple work-alike kit slammed with lawsuit by Apple.

A Los Angeles firm has tried to take a bite out of Apple Computer Inc.'s success by manufacturing a work-alike model of the Apple II microcomputer. But it's found this Apple can bite back.

Apple filed a lawsuit against Formula International for copyright, patent and trademark infringement.

The lawsuit, filed in U.S. District Court in Los Angeles, charges Formula with selling copies of the Apple II computer in kit form under the brand name "Pineapple." The kits are being imported from the Far East.

Kits include all components necessary for an Apple copy, including read-only memory containing Apple's copyrighted programs. Formula is also selling copies of Apple's copyrighted programs.

According to a statement from Apple, the lawsuit was filed to supplement actions by the U.S. Customs Service. That agency has been confiscating Apple copies being imported into the United States.

In recent months, Apple has taken extensive legal action against both American and Far East firms that it believes are infringing its proprietary product secrets. Actions have included conducting legal raids on companies in Taiwan. ■

Freud, Meet Dr. Micro

Buffalo psychiatrist uses Basic program to treat basic mental disorders.

A Buffalo, NY, psychiatrist has written a Basic program for the Apple II he claims can simulate the dynamics of human relationships.

The concept behind the program is called psychotronics. It uses a computer to allow a person to work out his or her problems in a programmed simulation.

Psychiatrist Ronald Levy, in a report appearing in *InfoWorld*, compares the idea to flight simulators used to train pilots:

"What it does is take certain kinds of interpretations and bring them to life in the form of a game that patients can play in which they can see things about their problems that they never saw before."

The program, Computerized Emulation of Personality and Environmental Conflicts, consists of a series of menus that offer a patient choices. The choices determine the next menu to appear on the CRT. They are also used to create a profile of the patient.

Before turning a patient over to CEPEC, Levy interviews the patient and assesses the areas in which he needs therapy. Different versions of the program have been tailored to deal with various neuroses, like depression.

The program analyzes simple neuroses by using medical stereotypes. Levy notes: "All medical reasoning is based on stereotypes. For example, no two schizophrenics are alike. (Psychiatrists) use a model or stereotype to help identify schizophrenia in a person."

He explained how the program and a therapist could handle a marital problem.

A wife (played by the computer) requests that her husband be neater around the house. A number of responses for the husband appear on the screen—accept the request, reject it, offer a counterproposal, abuse his wife. The husband makes a choice and another menu appears. Choices can be discussed by the therapist during the session or after the program's been run.

"It's not a game in the sense of someone winning or losing," Levy said. "What it does is allow people to take a fresh look at what they do in relationships."

So far, Levy has used the program only with patients he's known for a long time.

The Buffalo psychiatrist said the program was inspired by Wizardry, a game where players are offered a series of choices when confronted with monsters.

He claimed the program is more sophisticated than Eliza, a "psychotherapy" program developed as an exercise in artificial intelligence by Joseph Weizenbaum of the Massachusetts Institute of Technology. Rather than feed back what you give it, Levy explained, CEPEC assesses the patient to see how he or she responds to a situation.

He said, "The computer looks ahead to what you will do, based on what it knows about you from the choices you've made."

In practice, he continued, the program is closest to Cognitive Therapy, a program developed by Aaron Beck in the mid-1970s. That program focuses on people's underlying assumptions rather than their feelings.

Like most forms of therapy, Levy's program can only elicit an approximation of how people feel. "It's a garbage-in, garbage-out situation," he noted. "It's hard to profile a person who doesn't answer the questions honestly."■

Apple Tests Stimulation

Micro used to fathom mind's troubled waters.

Researchers in Providence, RI, are attempting to unlock the mysteries of mental health with an Apple computer.

The research team, led by Richard Haier, director of psychology at Butler Hospital in Rhode Island's capital city, uses the computer to control stimulation experiments on volunteer subjects.

Part of the research involves sitting volunteers in front of an Apple's CRT while the computer turns lights off and on in rapid succession. Electrodes attached to the person's head measure his or her response to the light's intensity.

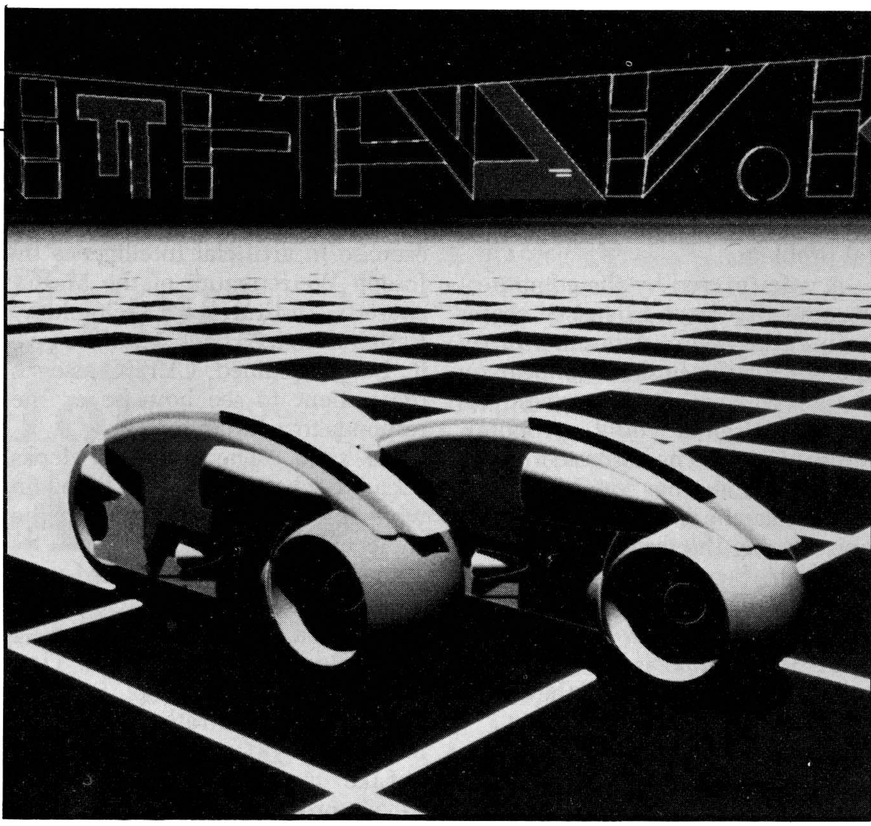
"There are individual differences in the way people tolerate stimulation," Haier told the *Providence Journal*. "Some people like a quiet weekend in the country, and some people find it excruciatingly boring. We think a large part of those differences comes from the way their individual nervous systems are wired."

"Schizophrenics are notoriously sensitive to stimulation," Haier said. "Depressed people are oblivious to stimulation. Manics crave stimulation."

By using the flashing lights of varying intensity, he explained, a computer can get a reading on whether a person is amplifying his or her experiences or blocking them out. That kind of information can help doctors diagnose patients.

Because the brain's "wiring" may play a role in the development of mental illness, the test may help doctors identify people who have a better than average chance of developing an emotional problem.

And because some portions of the brain waves might be used to measure how efficiently the brain processes information, Haier said the test may someday turn out to be the perfect way to measure someone's intelligence. "That idea is controversial," he noted, "but the data are very interesting."■



©1982 Walt Disney Productions

***TRON*: Apple effects are heard but not seen.**

TRON's Apple Sound

Sound effects expert says souped-up Apple better than some synthesizers.

Many sound effects in *TRON* were created with Atari and Apple microcomputers.

One of the key creators of the sound for the Walt Disney Productions film praised the micros in a story appearing in *Compute!* magazine.

"The amazing thing to me is the purity of the sound that comes out of the Atari, and also the Apple," observed Frank Serafine of Serafine FX Music/Sound Design located in Los Angeles.

"Their sound chips," he continued, "produce an extremely clean, pure sound which is even superior to some synthesizers I've worked with."

Unfortunately, he lamented, most people take sound effects for granted. The sounds in the film come and go so fast nobody realizes the amount of labor involved. He noted, "Several people spend a week of intense work to create something and it lasts only one second."

The light discuses used by the video warriors in the film are a good example of how much work went into each sound effect. Using an Apple II with plug-in sound cards from Mountain Hardware and an Alpha Centauri keyboard, Serafine overdubbed the micro's "bonging" sound with recordings of a bullwhip and screaming monkeys at the San Diego Zoo.

The Apple add-ons made the micro capable of a wider range of sound effects than the Atari, the SFX specialist commented. It also made the machine harder to use. But Serafine said the added capability was worth the extra effort.

He said the film's director, Steven Lisberger, demanded a concept for each sound effect.

He explained: "Like, for the disc-throwing sound, we came up with the concept that they had to sound beautiful, yet sad—sad because something so beautiful can at the same time kill. So overlaying the monkey screams lets you know that, although this flying disc is really beautiful, you'd also know you'd hate to be hit by it." ■

Apple Into IBM

A San Jose, CA, firm has announced a plug-in card for the Apple II allowing it to run IBM Personal Computer programs.

The card has a quasi-Intel 8088 microprocessor running at 5 MHz and 64K of random access memory. It costs $899.

The card is fully compatible with Apple peripherals and operates from the Apple's internal power supply. ■

UK Dealers Miffed

Apple's office in the United Kingdom has denied reports it favors dealers selling the Apples II and III over dealers selling only the II.

According to *MicroScope*, dealers in the UK have complained about Apple sending sales leads from advertising to dealers offering the full range of Apples rather than to the dealer nearest the lead.

"That's not the way we work," the fortnightly quoted an Apple spokesman as saying. "We routinely send inquiries out to the nearest outlet as a matter of routine." ■

by Paul Raymer

Scanner

Throughout this magazine—and most of the others you may read—you will find serious articles suggesting that your computer can do amazing number-crunching, alpha-reasoning, graphic-miracle things.

Of course it can. That's why you bought it! But, can you make it *do* those things? I never could.

I did have a wonderful time with my computer, though, just running simple programs that at least did something. As a natural klutz I never could get the two cross hairs to hit a space jerque, and to this day I find it hard to remember which comes first, the O or the H, in the formula for water.

Debit and credit, of course, are impossible.

I refuse to talk about Boolean logic at any time. I find it hard enough to spell.

Then I found out that if I just picked up a simple routine—from a magazine, from reviewing the Basic statements in a program, or from reading a "hint"—I could build a program around that routine. Finally, a thousand syntax errors later, I could get something to work—something I did—and it was all worth while.

Computing can be fun if you can make your machine do something. Join me. I'd like your viewpoint... and your program ideas.

For years I tried to develop a computer program to help my youngster learn how to spell. The project was difficult; no matter how much research I did at the blackjack tables, the roulette wheels or the crap layouts, I never was able to develop the right algorithm.

When my son was married recently, I decided that perhaps his interest in computer spelling might be diminishing. The heart of the program I'd been working on was called the ABC Finder, but, assuming the age level of my audience has risen, I have renamed it the Scanner.

You'll find it an easy program to type from a magazine into a computer because I have followed certain cardinal rules:

1. All of the characters in the listing are on your Apple keyboard.
2. Everything is neatly arranged.
3. Most lines are single statements. The program is easier to follow that way. Later on, if you wish, you can combine statements into one long line. Then your program will run faster and be more memory efficient. Big deal.
4. Ample remarks describe what's happening.
5. No profanity, poor grammar, hex or machine language is included.

"...it's really dandy how the little devils pop about the screen."

The game is fun—if you can ever get it to run. You get a chance to spell a bunch of words from a large selection of letters, and it's really dandy how the little devils pop about the screen.

Hint: Spell small words. You get more points, and will probably spell them more nearly correctly. You do not have to be a good speller to play this game. In fact you don't have to be a good speller to do much of anything any more. Sigh.

The Program

Here is the Scanner program! Don't fret, it's not as long as it appears to be, and most of the lines are short and easy to type. Warning: Watch out for line 210.

Line 100 clears the screen for action.

Lines 120–180 give well deserved credit to the author of this program.

Line 190 establishes the value of a variable. I read in a book somewhere that if you use the variable instead of the actual value, the program will run faster. I used it here because this is the number to make the buzzer work and it is such a darned long number, I usually get it wrong. N, I can get right.

Line 200 is a jazzy way of making a long dotted line. You could just say L$ = " = = = for 40 spaces" but this way is easier and neater. You may want to change its value, and with this technique you can. It is used here because periodically I like to show off this great dearth of knowledge I have!

Address correspondence to Paul Raymer, PO Box 42831, Las Vegas, NV 89116.

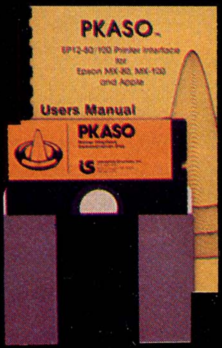

Nothing like it before. Nothing else like it now!
. . . brings you continuous Hi-Res action-animation in every adventurous moment! And, real running, leaping, crawling. Real fighting, shooting, stabbing, dynamiting. Real wounding, poisoning, killing. Real action, excitement, mystery! All in a real-time challenging adventure that's the wave of the future!
Paul Stevenson's graphic genius, first displayed in his best selling "Swashbuckler" sword fighting game, outdoes itself in AZTEC. You're inside an ancient Aztec pyramid searching for the golden idol. Descend deep into the heart of the temple—meet cobras, scorpions, giant lizards, hostile Aztec guardians and more. Watch for hidden trapdoors and strange death-rooms. Be ready to fight, or run, crawl or jump to possible safety. The menace is real, the options and strategy are yours. You've never seen an adventure like Aztec! You'll never tire of its amazing action-animation and exciting challenge. $39.95 for the Apple II* At your computer store or:
DATAMOST INC.
9748 Cozycroft Ave., Chatsworth, Ca 91311. (213) 709-1202.
AZTEC
VISA/MASTERCARD accepted. $2.00 shipping/handling charge. (California residents add 6½% sales tax.)
*Apple II is a trademark of Apple Computer, Inc.
Circle 36 on Reader Service card.

PKASO™

Printer Interface Family

A Masterpiece





PKASO Interfaces come complete with Cable, Instructional Diskette and Comprehensive Manual.

The PKASO family makes you and your Apple II or Apple /// a master of text and graphics.

PKASO makes it easy to use the features of your printer—select character sizes, vary line spacing, even print in colors. Simple PKASO commands make these features usable from the keyboard or a program.

PKASO also adds features to your system. Press a few keys and get a snapshot "dump" of the image you see on the screen—text or graphics. Add new characters and symbols that you couldn't print before, using our SuperFont™ system. Add our new PipeLine™ printing buffer and your printer can take its time while you and your Apple move on to the next task. The PipeLine is a modular add-on to the standard PKASO board.

The PKASO interface is designed for Apple II and Apple /// in all the popular configurations. It prints in full color on the IDS Prism Printer, and in striking black on C. Itoh, Centronics, Epson, IDS, NEC, and Okidata matrix printers.

Circle 28 on Reader Service card.

NEW!

The PKASO PipeLine™ printing buffer keeps your printer from tying up your computer.

- 8K to 64K Bytes of Memory.
- Micropressor controlled.
- Compression! Increases capacity to millions.
- Clear buffer and select modes easily.

is

Interactive Structures Inc.
146 Montgomery Avenue
Bala Cynwyd, PA 19004
Telephone: (215) 667-1713

Line 210 asks if you need to read instructions. If you don't, answer Y and the program goes on its merry way. You only get to see the rules if you insist. Had the same problem with my marriage certificate—but that is described in more detail in my book, "What do they mean when they say Double Occupancy in the Las Vegas Hotel commercials?"

Line 220 prints the title very slowly. I got the idea for this from a program sent in by Hartley G. Lesser (now on *inCider*'s staff) to the National User-Dump Group. I'm the chairman of their Nevada Chapter.

Lines 230–270 set up the alphabet display. Line 230 directs the program to create 39 letters. Then line 240 creates a random number. Line 250 adds 128 to the number so it can be poked into the right place. Line 260 makes noise (no wonder). Line 270 keeps the excitement going. Try this by not adding the number 128 in line 250—your program will look different. Exciting.

And it probably won't work.

Line 280 puts the cursor at the bottom of the screen, out of harm's way, and asks you to pick a letter. It then clears the line with CALL –868. It assigns a value of 1 to J, for a reason which escapes me now. I may think of it later.

Line 290 is a goof catcher. If the character selected is not a letter but a period, its ASCII number must be changed to a higher one; use 110. If you have one of those charts, you'll see that CHR$(110) is a lowercase n. Don't fret, I'll never use it that way.

Line 300 changes the value of the collected letters by adding 64 to them. Hey! That makes the n a period again when it becomes CHR$(174). Boy, I'm glad that's over.

Now scan the line.

Lines 320–340 look through the letters in the row. If a letter is found that matches your selection, then you've scored a hit. That is, if S (which is the value of the letter in the row) equals B (the value of the selected number, to which we added 64 to make it talk the same language to the poke/peek/thing) then the

***Program listing.** Scanner game in Applesoft Basic.*

```
100  TEXT : HOME : CLEAR
110  REM
120  REM

=======
SCANNER
=======

130  REM

     PAUL RAYMER

     PAUL'S ELECTRIC COMPUTER
     BOX 42831
     LAS VEGAS NEVADA 89116

140  REM  VI/XXVI/MCMLXXXII
150  REM
160  REM
170  REM
180  REM
190 N =  - 16336
200  FOR X = 1 TO 39:L$ = L$ + "=": NEXT X
210  VTAB 5: PRINT "IS THIS THE FIRST TIME YOU'VE PLAYED    'SCANNER'? (Y/
     N) ";: GET A$: HOME : IF A$ = "Y" THEN  GOSUB 820
220  SPEED= 39: VTAB 2: HTAB 15: PRINT "THE SCANNER": FOR Z = 1 TO 1000: NEXT
     Z: SPEED= 255
230  FOR X = 1 TO 39
240 R =  INT (26 *  RND (1)) + 1
250  POKE 1151 + X,R + 128
260 P =  PEEK (N)
270  NEXT X
280  VTAB 23: HTAB 2: CALL  - 868: PRINT "SELECT LETTER => ";: GET A$: VTAB
     23: HTAB 2: CALL  - 868:J = 1
290  IF A$ =  CHR$ (46) THEN A$ =  CHR$ (110)
300 B =  ASC (A$) + 64
310  REM

=========
SCAN LINE
=========

320  FOR X = 1 TO 40
330 S =  PEEK (1151 + X): IF S = B THEN  GOTO 360
340  NEXT X
350  GOTO 750
360  REM

=======
DROPPER
=======

370  VTAB 16: HTAB 1: PRINT L$
380  POKE 1151 + X,160
390  FOR Y = 3 TO 15
400  VTAB Y - 1: HTAB X: PRINT  CHR$ (32)
410  IF S = 174 THEN S = 110
420  VTAB Y: HTAB X: PRINT  CHR$ (S - 64)
430  FOR Z = 1 TO 20: NEXT Z
440 P =  PEEK (N)
450  NEXT Y
460  REM

=======
CATCHER
=======

470  FOR X = 1 TO 40
480 F =  PEEK (1831 + X)
490  IF F <  > 160 THEN I = F
500  NEXT X
510  REM

========
CLEAN-UP
========

520  FOR X = 40 TO 1 STEP  - 1
530  POKE 1831 + X,93: POKE 1831 + X + 1,160
540  FOR Z = 1 TO 30: NEXT Z
550 P =  PEEK (N)
560  NEXT X
570  POKE 1832,160
580  FOR X = 16 TO 19: VTAB X: HTAB 1: PRINT "]": VTAB X - 1: HTAB 1: PRINT
     " ": FOR Z = 1 TO 20: NEXT Z: NEXT X
590  POKE 1360,160
600  POKE 1488 + T,I:T = T + 1: IF T > 39 THEN  GOTO 760
610  REM

========
CLOSE-UP
========
```

Listing continued.

Listing continued.

```
620  FOR E = J TO 39
630 C =  PEEK (1151 + E)
640  IF C = 160 THEN 670
650  NEXT E
660  GOTO 280
670 H =  PEEK (1152 + E)
680  POKE 1151 + E,H
690  POKE 1152 + E,160
700  POKE 1191,174
710 J = E
720 P =  PEEK (N)
730  GOTO 620
740  END
750  REM

=======
COUNTER
=======

760  FOR X = 1 TO 40
770 U =  PEEK (1487 + X)
780  IF U = 174 THEN O = O + 1
790  NEXT X
800  VTAB 23: HTAB 1: CALL  - 868: PRINT "TOTAL WORDS = ";O;
810  END
820  REM

===========
HOW TO PLAY
===========
830  SPEED= 100: VTAB 5: PRINT "SPELL WORDS FROM THE ASSORTED LETTERS   DI
     SPLAYED...ONE LETTER AT A TIME --      WHEN ASKED.  WHEN YOU FINISH A
     WORD...   SELECT A <PERIOD>.": PRINT : PRINT "TO QUIT GAME...PRESS SPA
     CE BAR."
840  PRINT : PRINT "IF YOU SELECT A LETTER WHICH IS NOT IN  THE DISPLAY LI
     NE...THE GAME WILL END     AUTOMATICALLY."
850  SPEED= 255
860  VTAB 19: HTAB 1: PRINT "PRESS ANY KEY TO BEGIN GAME... "
870  VTAB 22: HTAB 1: PRINT "FOR SILENT VERSION PRESS <SPACE BAR>          =
     ====="
880  VTAB 19: HTAB 32: GET A$
890  IF A$ =  CHR$ (32) THEN N = 0
900  HOME
910  RETURN
```

program will go to line 360.

Line 350 comes up if no match was made, so the program skips all the neat stuff and goes to 750 where it has some routine chores to do.

Line 370 is that dotted line again.

Line 380 puts a blank space in the spot where the letter you selected was. It found it because line 330 can't keep a secret and so we know that the letter you selected was at X. Not any more. Now it is a blank space (ASCII 160 is a blank space in peek/poke talk).

Lines 390–450 are going to drop the letter down. Actually nothing drops. A blank space prints at Y–1. Then the letter prints at Y. This happens from 3 to 15—and the noise (line 440) makes it a thrilling adventure in computing.

After all this excitement you may want to stop a moment and observe that letters now appear at the top of the screen, and a letter just fell to near the bottom. Now what will happen?

Okay, so it's not *that* exciting! It *will* be when you put the catcher to work.

Lines 460–500 look along the line where the letters are supposed to fall, and, if one is there (really, if a location is not a space), then line 490 remembers that exact spot by giving I its value. More later on that bit.

Lines 510–600 tidy things up.

Starting at HTAB 40 you move from right to left to HTAB 1 (line 520) and as you do, you move the Mickey Mouse cursor across the bottom of the screen by printing a CHR$(93)—wonder what that is?—followed by a space so that it looks like it's moving. Each time we do it we make a noise (line 550). Small wonder!

Line 570 just puts the finishing touches on the mishmash started in line 530.

Line 580 pretty much does the same thing, vertically, and now the pseudo cursor moves from VTAB 16 down to 19. Can you feel the excitement mounting?

Use the same technique (printing a character followed by a space in the spot just vacated) only this time

SEE the characters instead of using ASCII numbers. No reason, just a change of pace.

Line 590 is a clean-up line similar in purpose to line 570. Must be a better way to do this, wouldn't you think?

Line 600 very grandly puts the letter originally selected into the right spot on the screen. The T+T+1 thing increments (Nevada high-class word for counts up) one spot so that the next letter will print in the right place and not cream this one.

Lines 610–650 close up the line of alphabet soup in the top line. Without this little handy dandy subroutine here, the letters at the top would look like the hero of Rocky IX with a big grin.

What this couple of lines is trying desperately to do is look along the line of ABCs to see if one of the letters is missing. If it is (line 640) then another bunch of lines does the work. If it doesn't find any spots with holes—make that ASCII 160—it just goes back to line 280 and starts its next little dance again.

If it does—aha—lines 670–730 go to work in a most industrious manner. It is all a little blurry as we look at it, but somehow each letter location is looked at and is moved to the left one space. This crunches the whole line down one space, which is really about all you can expect since you removed only one letter from the line. Stick a period at the end of the line because one needs to be there.

Do this routine again—and again—to move the entire line down. You need not start at 1 each time (oh, that's what the J in line 280 was for!), but can instead (very ingeniously) start where you left off. Makes for a much faster move, should you be anxious to get back to your checkbook program, or the complex instruction manual for the "Apple Mechanic." We make this move until the line is completely filled again and then continue at line 280, via the line 620 route. Whew!

Fortunately you're near the end of the program and lines 760–810 merely count the number of words created. Truthfully, this program isn't bright enough to count words—it can't spell worth a darn either—but it can count periods. Those are the little dots we had to use to indicate the end of a word. So, if you don't cheat when you play (and use all periods and no letters) the program will know how many words you made. Probably.

The game then, mercifully, ends at line 740.

Lines 820–910 are instructions for play which only need to be typed into the program if you actually intend to play the game of Scanner.

Scanner should prove the basis of a good, interesting, educational game that can be lots of fun. I suggest that, for the highest score, you keep words short and clean. Proper nouns, names of restricted drugs, TV stars and animals on the endangered species list should not be used.

If you have interesting ideas for adaptation of this program for business, commercial, military or gambling purposes, clip your idea to a crisp new dollar bill and mail to the author—care of most any Las Vegas casino. And quickly, please. ■

P.S. Throughout this program I refer to a variety of numbers from 1024 to 2039, the screen locations that can be accessed directly by peeking, poking and what have you. If you would like to know just where all those things are, send a self-addressed, business size, *stamped* envelope and I'll send you an Apple Text Graphic sheet which will show you where all those places are. It is assumed, of course, that your TV set or monitor is electrically compatible with mine here in Nevada. We use 60 cycles, 110 volts, AC and get the stuff directly from Hoover Dam. It may not work if your TV set uses Hertz or Avis or whatever.

Computer Metaphors

You don't need Apple Writer to play with words, but computers have added a few new twists.

by Tony Stavely

According to one theory, all thinking is metaphor. Language is full of it (e.g., snow *blankets* the ground). Very often, when problem solving, we seek analogies which make unfamiliar material seem like something we already know about.

In the history of psychology there are several instances of theorists using the latest human technology as a metaphor for the human mind or behavior. Descartes proposed that the body operated in a fashion similar to the hydraulically animated statues sometimes found in noblemen's gardens 350 years ago. Ever since the development of steam power we have applied a "boiler theory" of personality in which angry actions are called "letting off steam."

Fifty years ago our most complex device was a central telephone switchboard. Psychology textbooks then dutifully suggested that the brain was like a telephone switchboard connecting incoming calls (stimuli) with their intended receivers (responses).

Now, of course, we have computers. And many features of computing machinery and programming lend themselves to psychological metaphor. In fact, it works both ways. People commonly speak of what the machine "knows" or what it "thinks" as it executes a program. We say a subroutine is "looking for" the current value of a variable because it "wants" to multiply it by something and put it someplace.

Most of us who are involved with computers use this double-headed metaphoric thinking and find it valuable. During the past several months I have been paying particular attention to information processing that occurs outside of computers—in homes, offices and people's heads—and how the language of machine processing also applies there. What follows is a listing of some fundamental concepts in information processing and how they appear in everyday experience. The list is by no means exhaustive; make your own additions.

Buffer—a noun meaning an elastic storage place for information. Computers use buffers in places where one part of a process may proceed more swiftly than the next, and data need to be set aside during the catch-up period. Without a buffer there may be an overflow in which some information is lost.

Now, for years my desk was a mess. Each day's mail and papers descended on the pile left from preceding days. Usually I would remember the more important items, and I would be reminded of the pile's contents each time I rummaged through it looking for something.

Two months ago I began saying to myself, "I need an input buffer for my office." I figured that if I had an IN box and cleared it regularly, things wouldn't get lost or forgotten as often as they did. The little set of metal shelves for IN, IN PROCESS, and OUT that I got have helped a lot; my desk looks neater. I notice, though, that the first two shelves are almost ready to overflow.

Copy—to reproduce a set of information. Copying files, copying disks, making "hard copy" are all common computer activities. Very often information processing involves the literal movement of information from one place to another. Communication involves sending copies to destinations while retaining originals for future reference.

Historical progress consists of a series of improvements in our ability to copy information. Consider this series: clay tablets, papyrus, quill pens, movable type, typewriters, phonograph records, linotype machines, cassette tape recorders, throw-away ballpoint pens, Xerox.

Before hard copies there was only word-of-mouth. Information had to be memorized and transported in the heads of priests and balladeers. Thanks to that original form of copying we still

Address correspondence to Dr. Tony Stavely, Psychology Department, Keene State College, Keene, NH 03431.

pple II is a registered trademark of Apple Computer Inc.
oftcard is a registered trademark of Microsoft Inc.
P/M is a registered trademark of Digital Research Inc.



ORANGE 16K
Expands APPLE II® usage from 48K to 64K. Functionally equivalent to the Apple Language card. Provides 56K CP/M® with Microsoft's SoftCard®.

ORANGE 80
An 80 column by 24 line video display card for the APPLE II®. Compatible with Integer BASIC, Applesoft, Pascal and CP/M®. Features include: inverse and flashing video. Operates with monitors and most home black and white television sets.

ORANGE KEYBOARD ENCODER
Provides true shift key operation for upper and lower case entry with auto repeat on every key. 60 programmed and 5 user programable function keys.

ORANGE ROMFONT BANK
Enables the APPLE II® to simultaneously generate and mix any 3 typefaces on the text screen. 26 different ROMFONT typeface chips available.

ORANGE ROMFONTS
Text screen character generator chips for APPLE II® computers. 26 typefaces available. All in full upper and lower case.

ORANGE PROTO-BOARD
A prototyping card for the APPLE II®. Ideal for the hobbyist and hardware developer.

ORANGE PAIR
A Z-80® card for the Apple II. Includes 64K of on board RAM. Multi-tasking with Apple's 6502 and 48K RAM allows I/O spooling with the keyboard, printer and disk. Compatible with Microsoft Softcard diskettes. Comes with CP/M.

See your local ORANGE dealer today.

orange computers inc

orange computers inc.
10-1495 bonhill road
mississauga, ontario
canada L5T IM2

Available in Canada, USA, Hong Kong, Singapore, Australia and Japan.

Circle 153 on Reader Service card.

have the *Iliad* and the Book of Genesis.

Database Management—this weighty phrase refers to various aspects of the care and feeding of collections of information. DBM activities include the storing of information in an organized "structure" of some sort, sorting sets of items into categories, searching collections of stored information for particular items—perhaps with the help of indexes or catalogs—and revising, updating or purging parts of a collection.

"Library" is an older but perfectly respectable name for "database." The latter term probably results from the heavy concentration of numerical information in early computer applications. The principles of database management are approximately the same whether the material stored is binary numbers, books, magazines, recordings or auto parts. Many of us do a poor job of managing our personal databases, as indicated by the amount of search and retrieval activity needed at income tax time.

Default—the thing that happens when nothing is done; often used with reference to computer programs as the "default conditions" and meaning the behavior of a program when a user presses the return key without typing any data in.

In everyday life the concept applies best to committees and other organizations. A common scenario begins with a meeting in which decisions are made and actions are planned with responsibilities delegated to various members. Often, the members fail to perform their duties—they default—and the planned projects do not materialize.

Computer programmers soon learn to mind their default conditions, especially if unsophisticated users are expected to run their programs. Otherwise the computer may get lost in an infinite loop or sit waiting for an unprompted input. Groups and organizations, not to mention individuals, must also mind their default conditions if they are to be successful.

Procedure—this is the Logo programming language name for a program or subroutine. Non-computer names for "procedure" are "recipe" and "the instructions" (as in, "Oh dammit! I wish I had read the instructions first.") Anyone who has written a computer procedure is aware of how literal the machine is. It does what you tell it and only that, unless it has already been programmed to make guesses and draw inferences from partial information.

In human activities many procedures are not explicitly stated. For example, the procedure for tying a bow knot is well-mastered by most of us but implicitly learned. To demonstrate, write a set of instructions for tying your shoes and then try to follow it exactly. You are likely to encounter one or more "bugs" in the stated procedure.

When a computer program fails, the programmer knows that at least one instruction in it is incorrect or missing. He or she debugs the procedure to root out the error. When plans do not work out in everyday affairs,

''Library'' is an older but perfectly respectable name for ''database.''

you can take a similar approach. Initiate debugging activities to find where the procedure went wrong. Unfortunately, we try more often to fix blame on a person than to fix the procedure itself.

Representation—in artificial intelligence this term refers to the form in which a problem is entered as data for a problem-solving computer program. Non-computer problem solvers say more or less the same thing: How you state a problem has a lot to do with being able to solve it.

"Representation" applies to computing in another way also. Deep down, the computer is a simple device that adds numbers. The numbers are *represented* as strings of zeros and ones. At the user's level the machine may be doing word processing, drawing pictures or piloting the Enterprise; but in the central processing unit it's all lightning movement of bits of electric charge.

The lesson to draw is that the same information can be represented in many ways—numbers, words, pictures, sounds, gestures, etc. If you learn enough about the different modes of representing a set of information, you'll recognize the sameness beneath the surface. When Descartes was not examining the clockwork robots of his day, he was inventing ways to translate algebraic expressions into geometric figures, and vice versa.

Transform—to change something into something else. Usually, transformation involves the application of a rule to elements of a set of items of information. One such transformation is to square each number on a list; another is to convert every sentence in the passive voice to active voice in a draft of an essay. Procedures perform transformations.

There is no limit to the procedures we can devise and so there is no limit to the changes we can make. New things grow out of old. Sometimes it is hard to backtrack the rules being applied and understand how we got here from there. Because human procedures are often not explicit in the first place, the difficulty is compounded.

Zoom—not a computer term at all, "zoom" refers to the action of a zoom lens on a camera. As an operation performed on information, I use "zoom" to stand for the possibility of changing the scale of representation, making it larger or smaller. We can magnify the information by extracting a subset and adding detail to it. Alternatively, we can summarize the information set by removing detail and retaining the essentials, as in writing a précis of some longer work.

The history of science and technology is studded with inventions to increase our ability to magnify or summarize information of one sort or another. Only after we could "summarize" the planet with a photograph from space did the concept of Spaceship Earth become popular.

Perhaps it is no great advance to call an in-basket an input buffer or to refer to one's personal financial records as a database. However, since I began speaking this way about everyday activities I have come to understand better what I have been doing all along (and needed to debug). ■

Out of the Woodwork...

All was bedlam... until The Seer made way into the Valley.

by M. Duff

Dusk, tainted red, settled upon The Valley. High atop a cliff that overlooked the warring factions, The Seer sat. With a weary sigh, he turned to those who had for so long maintained The Orchard. Only now they watched in fear as the fruits of their labors crumbled into bits and pieces, while the enemy nibbled ever closer to the core. The question as to why they had waited so long to call upon him could be answered by the pride that had fallen from their faces.

"Sir, we come to you for advice," stated the one designated as Leader. "The war no longer wears well. For four years our entry has remained unchanged. But now the ravages of time and competition have weakened the warp and woof of our mainstay. What is to become of us?"

"And that is not all," added a smartly attired nobleman. "We now suffer attacks from within! The enemy uses materials created strictly for us. The legality of such an affront must be decided again. Frankly, we've more than enough worries concerning the front. Must we bother still further with these upstarts?"

"An ace is played whenever held," answered The Seer. "Mimicry is often the basis of success."

"To use our own processes against us is unthinkable," the nobleman said, a whine edging his words. "Our power has been blunted. What will our good name be worth now?"

The Seer folded his hands and looked upon the VIPs assembled beneath the rainbow-hued banner.

"The complacency of days past is now a luxury you can ill afford. Were you not warned? Look upon the designs of those who would be kings."

A sweeping hand indicated The Valley below.

"To the east, does not the sony rise ever higher upon the horizon? One suffers a sore nec from ever watching canon fire light up the skies.

"Yet, is there not perhaps even more to fear from the opposite direction? Watch, for your lower end is another front that flirts with disaster. Your fleet is under assault by an enemy commodore and will sinclair to the bottom unless new strategies are enacted immediately.

"From the south a tandy array encircles beleaguered forces, while the zenith of accomplishments halos those in baby blue. The north star shines brightly as combined forces vector near Texas."

"What are we to do?" asked one old warrior. "The peripherals stemmed the attack before. Why not again?"

"And bally well you should ask," rejoined The Seer. "Four hundred—nay, eight hundred—sapped our strength when they invaded our territory. Look upon the victory of the twenty in the common marketplace. Is this not likened to an eagle in flight, which at first battles to gain the heights, only to soar effortlessly upon the winds?"

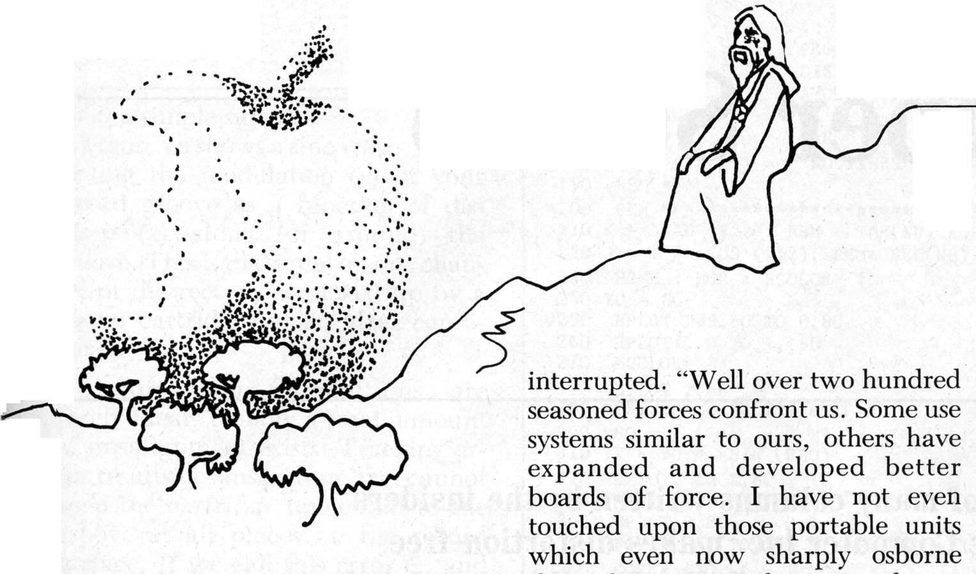
The directors shifted uneasily and murmured among themselves. "We started out well, maintained our strength with a dependable corps of engineers, cultivated an enormous base of users... how can we fail now?"

"The chip on your shoulders is weak," replied The Seer. "Those Z-80 and Z-80A machines are bent upon universal acceptance, never mind the tanks of the 6509s and 6510s from across the sea. And certainly the entrenching of the 8085As, 8086s and 8088s will have enormous impact upon our forces.

"Our eight-inch shells continue to hold sway, but the sixteen-inch shells are far more effective. And our concern need be genuine when the thirty-two-inch shells begin descending upon us."

The silence was deafening. Pervasive gloom shrouded the encamp-

M. Duff is an exceedingly private person of unknown age and gender. We have no idea where Duff lives, or whether we'll hear from this author again. Send correspondence c/o inCider, *and we'll leave it on the kitchen table for him/her.*



ment. The Seer nodded his head.

"You must not allow success to run as sand through your fingers. Think upon those whose lives depend upon your existence. Over five hundred thousand loyal users. An equally impressive number of tradesmen distributing compatible wares to your forces. Where are they to turn should you fail?"

"Now, as lobo the wolf nips at our heels, the realization comes that the time for game-playing is over," spake the chief orchardist as he stepped forward. "We have our reserves, Seer. Not simply mirror-images of our foes, but hardware that shall make all stand up and take notice."

"For instance?" inquired The Seer.

"Forces named II-E shall arrive shortly to do battle, to remove the impact of aliens from within our academies. Battering rams, sixty-four strong, will restore our status as leader."

"Anything else?"

"Those in business shall clamor for our lissome Lisa when she appears. Infant red, baby blue, it matters not. All shall reel from Lisa's introduction."

The assemblage waited for the impact the chief's words would have upon The Seer. He merely laughed.

"Talk is cheap. Gossip holds little attention after a tongue has wagged for so long. You may very well have a Lisa, and a II-E, but shadows perform poorly. To stem the tide—nay, the flood—of lower-end systems, you have naught."

"The cost of development, the times of. . ."

"Listen, and listen well," The Seer interrupted. "Well over two hundred seasoned forces confront us. Some use systems similar to ours, others have expanded and developed better boards of force. I have not even touched upon those portable units which even now sharply osborne themselves into our laps. He who ignores what encompasses him is either a fool with more capital than cents, or one who thrives upon Chapter Thirteen.

"The original errors committed with the III should have been an education, not a stone wall. Open your eyes. What once was, and remains, a power, need not be reduced to impotence by imports or domestics."

"It's too late," moaned a youthful member of the group.

"No," snapped The Seer. "Show those who support you your resilience. Develop machines capable of competing upon *all* levels. Don't alienate those who trust you. But most of all, recapture the pioneer spirit that gave you birth."

The Seer rose to leave.

"But wait," cried one of the youngest in the gathering. "What is your advice? That we merely regroup?"

With a smile The Seer strode towards the speaker and laid a hand on his shoulder.

"Many have spoken before me, and many will speak after I have gone. Liken my words to the perpetual drumming of rain upon a roof. Sooner or later, some of it will leak in."

And he was gone. ■

Super Stereo

This first of many columns written by the insiders at Apple Computer Inc. makes distortion-free stereo a possibility for all.

by Phil Roybal

Does your stereo system sound like what you paid for it?

You may have lavished funds on elegant electronics, only to feed them a poor-quality signal and wonder at the miserable sound that resulted. Since you can encounter substantial distortion right where the needle meets the record, it pays to make sure that the phono cartridge source is adjusted for optimum performance. Your Apple II can help you do it right.

Distortion is unavoidable in normal record reproduction, even with a good-quality cartridge and a light, modern tone arm. Why? Aside from electrical limitations of the components themselves, no pivoted tone arm can accurately track across a record's surface. It moves its cartridge in a curve, and proper tracking actually requires a straight-line motion. But proper setup of the cartridge can bring this tracking distortion within acceptable limits, actually reducing it to zero at some points on the record's surface. This article tells how to get there.

The Essence of the Problem

Let's start by using the Apple to model the effect of tracking distortion

Address correspondence to Phil Roybal, European Marketing Manager, Apple Computer Inc., 10260 Bandley Drive, Cupertino, CA 95014.

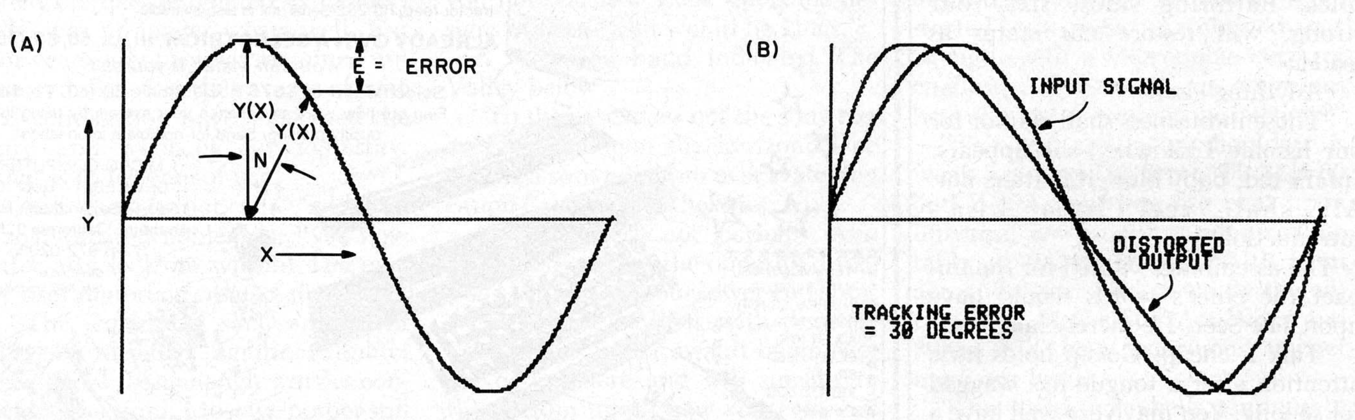


Figure 1. Tracking Distortion for Sinusoids. Figure 1a shows a sine wave recorded signal whose amplitude, y(x), is a function of position (x) along the record groove. Tone arm tracking error has the effect of "tilting" the cartridge forward in time, so that it picks up the signal y'(x) instead of y(x). The difference, ∈, between these two signals is the error. Figure 1b compares a sine wave input with the distorted output which would result from a tracking error of 30 degrees.

Program listing 1.
A Tracking Distortion Graphing Program.

```
100  REM ***********************************************************
110  REM *                                                         *
120  REM *                    DISTORTION GRAPH                     *
130  REM *           PROGRAM TO CALCULATE AND PLOT THE             *
140  REM *           DISTORTION RESULTING FROM A GIVEN             *
150  REM *           AMOUNT OF TONE ARM TRACKING ERROR             *
160  REM *                                                         *
170  REM *           (REQUIRES HI-RES CHARACTER GENERATION         *
180  REM *            ROUTINES FROM THE DOS TOOLKIT.)              *
190  REM *                                                         *
200  REM ***********************************************************
210 K =  TAN (.52): REM   TAN(NU) FOR NU = 30 DEGREES
220 J = 1 /  COS (.52): REM  SEC(NU)
230  HOME : HGR : HCOLOR= 1
240 YO = 80
250  HPLOT 279,80 TO 0,80
260  HPLOT 1,0 TO 1,160
270  HCOLOR= 3
280  HPLOT 2,80
290  FOR X = 3 TO 277 STEP 2
300 RHO = X / 44
310 YX = YO *  SIN (RHO)
320  HPLOT  TO X,80 - YX
330  NEXT X
340  HPLOT 2,80
350  FOR X = 2 TO 279 STEP 4:RHO = X / 44
360  GOSUB 470
370  HPLOT  TO X,80 - ZX
380  NEXT X
390  VTAB 23: PRINT "  TRACKING DISTORTION FOR SINUSOIDS"
400  VTAB 3: HTAB 17: PRINT "INPUT SIGNAL"
410  VTAB 12: HTAB 24: PRINT "DISTORTED": VTAB 13: HTAB 26: PRINT "OUTPUT
     "
420  VTAB 16: HTAB 3: PRINT "TRACKING ERROR"
430  VTAB 17: HTAB 4: PRINT "= 30 DEGREES"
440  GOTO 440
450  REM


        CALCULATE Y(RHO+DELTA RHO)
460  REM  DELTA RHO=K*SIN(RHO+DELTA RHO)
470  IF  SIN (RHO) < 0 THEN 540
480  FOR TR1 = 0 TO .6 STEP .01
490 ERR = TR1 - (K *  SIN (RHO + TR1))
500  IF  ABS (ERR) < .01 THEN 520
510  NEXT TR1
520 ZX = YO *  SIN (RHO + TR1)
530  RETURN
540  FOR TR1 = 0 TO  - .6 STEP  - .01
550 ERR = TR1 - (K *  SIN (RHO + TR1))
560  IF  ABS (ERR) < .01 THEN 580
570  NEXT TR1
580 ZX = YO *  SIN (RHO + TR1)
590  RETURN
```

This program will produce a graphic image on the hi-res screen of a sine wave input and the distorted output resulting from any specified angle of tracking error.

upon a simple signal.

Figure 1a shows a sine wave, representing the modulation (y) of your record groove as a function of distance (x) along (or around) the groove. This is the signal on one channel of the record, as picked up by a phono cartridge under ideal conditions.

Unfortunately, conditions are rarely ideal. Usually a small amount of misalignment exists: Tracking error results because a tone arm cannot keep the cartridge tangential to the groove at all places on the record surface. If we call this error $\in$, and show it graphically as a "tilting forward" of the point, y(x), where we pick up the sine wave we see that the picked-up waveform develops a distortion proportional to the amplitude of the recorded signal at each point. This effect is equivalent to a compression of the time that it takes for the signal to swing positive, and an extension of the time that it takes for the signal to swing negative. It acts to introduce into the picked-up signal some irritating harmonics of the original sine wave.

Figure 1b shows the initial sine wave and the distortion resulting from a gross cartridge misalignment of 30 degrees. (Of course, such drastic misalignment would be rare, but it allows the effect to be easily seen.) Listing 1 shows an Applesoft Basic program that will generate these curves for any given amount of tracking error. It was used, along with Apple's new DOS Toolkit (for hi-res text), for producing Figure 1b.

Approaching a Solution

Two types of tone arms are in common use today: straight and offset designs. A brief examination of how they traverse a record's surface shows that they have slightly different tracking error characteristics. As a matter of fact, if you construct the equations for the two different arm geometries, you'll find that the errors approximate the results shown graphically in Figure 2. The straight arm, with one degree of design freedom (pivot angle), has a tracking error expressed by graph T1. The offset arm, sporting two degrees of design freedom (pivot angle and offset angle), more closely approximates zero tracking error as shown in graph T2.

A little examination of Figure 2 reveals an item of interest: A properly set-up tone arm can produce zero tracking error in at least one place on the record. For straight arms, that zero point will occur somewhere near the center of the recorded area, and absolute tracking error will rise to some maximum value at the extremes of the recorded band.

Offset arms yield a bonus that compensates us for their extra weight: zero tracking error at *two* places on the record. At other locations, distortion will rise to a maximum value that's only half that produced by a straight arm.

Putting It Into Practice

This may seem interesting but a little abstract. Straight arm or curved, you're probably content with what you've got, and knowing its distortion characteristics won't make it sound any different. But these graphs express ideal conditions, and most phono cartridges don't operate under

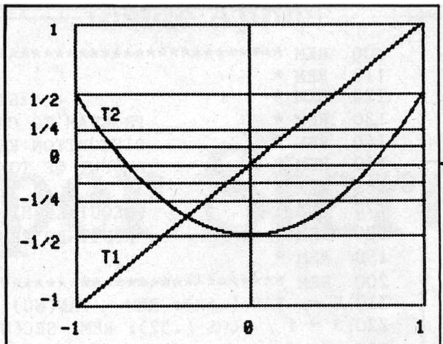


Figure 2. Tracking Error.

This figure shows normalized tracking error between a record's inner and outer grooves for straight (T1) and offset (T2) tone arm designs. Note that the offset design allows lower overall error, and actually produces zero error at two points on the record surface.

ideal conditions. Why not? Because we don't mount the cartridge (or the tone arm) properly, even though we may follow the manufacturer's instructions. As a result, cartridge overhang (Figure 3) is incorrect. Here's where you (and the Apple) can help.

If you were to mathematically model the action of your tone arm crossing a record surface, you could derive the equations for optimal over- or underhang (the distance between the tip of the stylus and the center of the record spindle) shown below. These presume an arm designed for 33⅓ RPM records, and mounted to provide the minimum (optimal) tracking error over the record surface.

Straight Arm Optimal Underhang (stylus distance short of the spindle center):

$$U = \frac{r_1^2 r_2^2}{L(r_1^2 + r_2^2)}$$

Where r_1 and r_2 stand for the inner and outer limits of the recorded band, and equal 146 and 60.3 mm respectively. L = Effective Tone Arm Length.

Offset Arm Optimal Overhang (stylus distance beyond the spindle center):

$$O = \frac{r_2 r_1}{L\left[\frac{\left(\frac{r_2 + r_1}{2}\right)^2}{r_2 r_1} + 1\right]}$$

Running these equations out on your Apple provides the cartridge positioning table in Table 1. With the program in Listing 2, you can produce this table for almost any size tone arm. Then, having computed the desired overhang/underhang, you just establish that adjustment. Here's how.

Looking straight down at your record player, locate the center of your tone arm's bearing rotation (the pivot point of the arm on its pillar) as best you can. Mark it with a pencil. Then measure from there to a point directly over the tip of the stylus. This is your arm's effective length, L (Figure 3). Run the program in Listing 2 using this length to compute your desired overhang/underhang.

EFFECT. LENGTH (MM)	STRAIGHT UNDRHANG (MM)	OFFSET OVHANG (MM)	OFFSET ANGLE (DEG.)
260	12	-15.3	21.1
261	11.9	-15.3	21
262	11.9	-15.2	20.9
263	11.8	-15.2	20.8
264	11.8	-15.1	20.7
265	11.7	-15.1	20.7
266	11.7	-15	20.6
267	11.6	-14.9	20.5
268	11.6	-14.9	20.4
269	11.6	-14.8	20.3
270	11.5	-14.8	20.3
271	11.5	-14.7	20.2
272	11.4	-14.7	20.1
273	11.4	-14.6	20
274	11.3	-14.6	20
275	11.3	-14.5	19.9

Table 1. Table of Underhang/Overhang Values. A properly set up tone arm will produce the underhang or overhang shown here. To find the proper value for other arm lengths, just run the program shown in Listing 2.

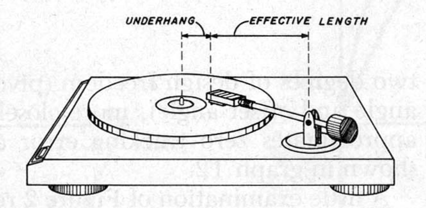


Figure 3. The Physical Meaning of Underhang and Effective Length. Effective length is the distance between the tone arm's vertical pivot axis and the stylus point. Underhang is the minimum distance between the stylus point and the center of the spindle. A small variation here can make a big difference in distortion levels.



Setting Up the Tone Arm. The easiest way to set up your system is to make a simple protractor, as shown here. It will help you locate the zero-error points and accurately align your cartridge.

Now, loosen the cartridge mounting and gently maneuver the cartridge until you have established the specified overhang (beyond the spindle center) or underhang (short of the spindle center). (If you have difficulty measuring to the center of the spindle, measure to its side and add 3.6 mm.) Then tighten the cartridge mounting bolts *slightly*, so the cartridge will not slip out of place but you can still twist it. Note: Some arms feature positioning screws that let you make this adjustment without loosening the cartridge mounting.

Recheck your effective length (L). In the unlikely event that you've had to make a large adjustment to cartridge position, L will have changed, and the desired overhang with it. You must sneak up on the overhang recursively. (Obviously, it is possible that the arm may be mounted in such a position that correct overhang cannot be achieved at any cartridge adjustment. This spells hard choices: permanent distortion or remounting of your arm.)

Now, with the cartridge almost in place, the next step is to twist it in its mounts to get zero tracking error at its one (straight arm) or two (offset arm) optimum point(s). For that you'll need a special homemade tool.

Build yourself a little protractor using a 4 × 6 inch index card. Mark a spindle point in one corner, and draw a line from that mark diagonally across the card. Along that line, mark the following distances from the center of the spindle mark:

Straight Arm: 66.0 mm
Offset Arm: 66.0 mm and 120.9 mm

These are the zero-tracking-error points for the two different arm geometries.

Now, using a square, make a series of closely-spaced, parallel lines perpendicular to this diagonal line across the card. Center these group(s) of lines on the mark(s) you made above, making sure that they extend beyond the marks on each side by a little more than ½ the width of your phono cartridge (see the photo).

Now punch a hole in your card at the spindle mark, and mount the card on your turntable platter. Reduce the antiskating bias on your tone arm to zero, and carefully manipulate the platter and tone arm until you can set the stylus precisely on one of your marks. Look at the cartridge, head on, in a strong light. Its sides should line up with your protractor's parallel lines. If they don't, gently twist the cartridge in its mounting until they do. For offset arms, you can now move the stylus to the second zero-error point to check your alignment. Tighten the cartridge in its mounting, being careful not to disturb its alignment. Then recheck it for good measure.

Now you're ready for final setup. Get out your stylus pressure gauge and rebalance your tone arm for the tracking force recommended by the cartridge manufacturer. (Stylus pressure may have changed due to remounting of the cartridge.)

Finally, restore the antiskating force to correspond to the tracking weight you're using. Then turn off your Apple and enjoy the music! ■

Bibliography

H. J. Baerwald, "Analytic Treatment of Tracking Error and Notes on Optimal Pick-Up Design," *J. Soc. Mot. Pict. Eng.*, Dec. 1941, p. 591.

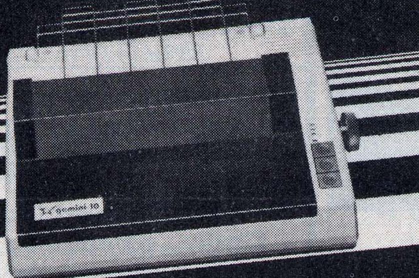
P. Aczel, "Cartridge/Arm/Turntable Follow-Up: Loose Ends and New Developments," *Audio Critic Mag.*, Vol. 1 No. 6, Spring 1978, p. 43.

Applesoft Basic Programming Reference Manual, Apple Computer Inc., 1978.


```
120  REM **************************************************
130  REM *                                                *
140  REM *                OVERHANG TABLE                  *
150  REM *        PROGRAM TO CALCULATE THE REQUIRED       *
160  REM *        TONEARM OVERHANG AND OFFSET ANGLE       *
170  REM *        FOR MINIMUM DISTORTION                  *
180  REM *                                                *
190  REM **************************************************
200  TEXT
210  HOME : VTAB (10)
220  INPUT "START WITH WHAT LENGTH? ";MN
230  HOME
240 R1 = 60.325:R2 = 146.05
250  PRINT : PRINT "-----------------------------------"
260  PRINT "EFFECT.  STRAIGHT  OFFSET   OFFSET"
270  PRINT "LENGTH   UNDRHANG  OVHANG   ANGLE"
280  PRINT " (MM)      (MM)     (MM)    (DEG.)"
290  PRINT "-----------------------------------"
300  FOR L = MN TO MN + 15
310 PROD = R1 * R2:SUM = R1 + R2
320 S1 = R1 © 2:S2 = R2 © 2
330 SO = S1 * S2 / (L * (S1 + S2))
340 OO =  - (PROD / L) / (((SUM / 2) © 2 / PROD) + 1)
350 O1 = (SUM / L) / (((SUM / 2) © 2 / PROD) + 1)
360 OA = 360 * (1 / 6.28) * ( ATN (O1 /  SQR ( - O1 * O1 + 1)))
370 B = ( INT (10 * SO + .5) / 10)
380 C =  INT (10 * OO + .5) / 10
390 D =  INT (10 * OA + .5) / 10
400  PRINT  TAB( 2);L; TAB( 11);B; TAB( 20);C; TAB( 31);D
410  NEXT L
```

Program listing 2. Overhang Table Program. This program will calculate underhand or overhang for any tone arm size, using the equations described in the text.

Star Micronics **GEMINI-10**



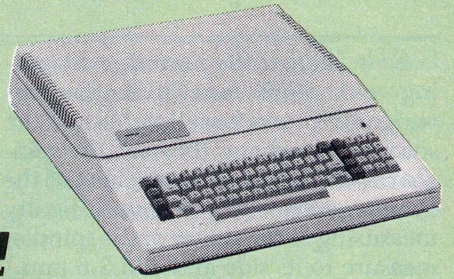
$419.88 UPS DELIVERED

- 100 characters per second, bi-directional, logic-seeking printhead action (48 lines/min.) with 2K print buffer, expanable to 4K on-board
- 9 x 9 matrix produces proportional, 10, 12, 17 cpi with true descenders, plus double width, double strike, italics, & special graphics characters
- 120 x 144 hi-resolution dot-addressable graphics matrix
- Subscripts, superscripts, underlining, backspace, plus 2K user-programmable character ROM, perf skip, vert/horz tabs
- Friction/tractor standard; handles 3-part forms (8.5")

The Franklin **ACE 1000**

An Apple-compatible Personal Computer

FRANKLIN
COMPUTER CORPORATION



The Franklin **ACE 1000** includes 64K RAM, upper/lower case character set (40 columns), hi-res graphics, numeric keypad, 8 peripheral slots, joystick/paddle connectors, and a built-in cooling fan. The Franklin **ACE 1000** has full hardware/software compatability with all existing Apple II® products.

Our special package price includes The Franklin **ACE 1000**, one **ACE 10** disk drive with a drive controller card, and the **PI-4** 9" Amber Monitor and video cable.

All for only **$1549.64**, UPS delivered.

PRINTERS

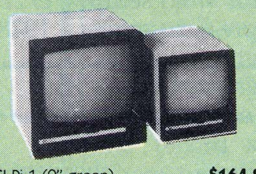
Anadex DP-9500A	$1459.88
Anadex DP-9501A	$1459.88
Anadex DP-9629A	$1549.88
Centronics 122-1	$829.88
Centronics 122-3	$949.88
Centronics 352	$1649.88
Centronics 353	$2324.88
C. Itoh Prowriter	$499.88
w/RS-232C	$609.88
C. Itoh Prowriter 2	$734.88
w/RS-232C	$789.88
C.Itoh F-10 Starwriter, 40 cps Parallel or RS-232C	$1499.88
C.Itoh F-10 Printmaster, 55 cps Parallel or RS-232C	$1799.88
F-10 Tractor	$289.88
Daisywriter 2000	$1089.88
Daisywriter Tractor	$149.88
Daisywriter Cable	$49.88
Diablo 620	$1269.88
Diablo 630	$1969.88
Diablo 630 KSR	$2694.88
630 Tractor	$314.88
DMP-85 Printer	$469.88
IDS Microprism	$679.88
IDS Prism 80	$1104.88
Prism 80 w/graphics	$1339.88
Prism 80 w/sheetfeed	$1459.88
Prism 80 w/4-color	$1539.88
IDS Prism 132	$1269.88
Prism 132 w/graphics	$1339.88
Prism 132 w/sheetfeed	$1459.88
Prism 132 w/4-color	$1699.88
Microline 80	$349.88
Microline 82A	$439.88
80/82A Tractor	$59.88
82A Roll Paper Holder	$49.88
Microline 83A	$694.88
82A/83A Okigraph ROM	$44.88
Microline 84 w/graphics & tractor Parallel, 200 cps	$1044.88
RS-232C, 200 cps	$1164.88
NEC PC-8023A	$509.88
NEC 3510	$1929.88
NEC 3530	$1809.88
NEC 3550	$2199.88
3500 Tractor	$239.88

PRINTERS

Qume Sprint 9/45	$2109.88
Smith Corona TP-1 10 or 12 cpi, parallel or RS-232C	$599.88

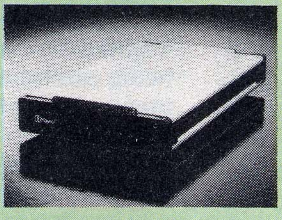
CALL FOR PRICES on Epson, DIP, MPI, Datasouth, & other printers.

MONITORS



USI Pi-1 (9" green)	$164.88
USI Pi-2 (12" green)	$174.88
USI Pi-3 (12" amber)	$199.88
USI Pi-4 (9" amber)	$169.88
Amdek 300G (12" green)	$174.88
Amdek 300A (12" amber)	$189.88
Amdek 310G (12" green)	$209.88
Amdek 310A (12" amber)	$209.88
Amdek Color I	$379.88
Amdek Color II (Hi-res RGB)	$799.88
Amdek Color III (RGB)	$499.88
Apple RGB Interface	$164.88
Electrohome 1 (RGB)	$339.88
NTSC (RGB option)	$84.88
Electrohome 2 (Hi-res RGB)	$594.88
Super Color Option	$214.88
NEC (12" RGB)	$759.88
NEC (12" Color)	$364.88
Princeton Graphic Systems HX-12 (Hi-res RGB)	$699.88
Zenith TVM 121 (12" green)	$139.88

MODEMS



Hayes Smartmodems feature programability in any language, auto dial & auto answer, full & half dup ex, keyboard control, user-selectable prameters, 7 LED status lights & an audio monitor. Comes complete with power supply and modular telephone cable.

Hayes 300 Baud	$239.88
Hayes 1200 Baud	$569.88
Hayes Micromodem-II	$299.88
w/Software	$329.88
Novation Apple Cat 1200	$579.88
Novation Apple Cat 300	$334.88
Novation 1200 Upgrade	$324.88
Novation Auto Cat 300	$224.88
Novation Auto-Cat 1200	$569.88
Novation Cat	$159.88
Novation D Cat	$159.88
Signalman Mark 1	$89.88
Multi-Line Junction Box	$39.88

APPLE PERIPHERALS

PKASO Smart Printer Interfaces
For Apple/Franklin centronics parallel output. Specify C.Itoh, Centronics, Epson, IDS*, NEC or Okidata printers.

PKASO card & cable	$139.88
*IDS Color card & cable	$159.88

APPLE PERIPHERALS

Parallel Card w/cable	$89.88
Corona Data Systems	
5mB Hard Disk	$2049.88
10mB Hard Disk	$2459.88
MicroSoft	
Premium Package—A 16K RAMCard, Z80 SoftCard, Videx VideoTerm w/soft video switch, & CP/M	
Users Guide	$519.88
Z80 SoftCard	$269.88
16K RAMcard	$139.88
Mountain Computer	
5mB Hard Disk	$2549.88
10mB Hard Disk	$2939.88
Card Reader	$1399.88
CPS MultiFunction	$174.88
A/D—D/A Convertor	$314.88
RAMplus 16K+ RAMcard	$164.88
RAMplus 32K+ RAMcard	$189.88
ROMplus +	$144.88
ROMwriter Card	$159.88
SuperTalker	$179.88
Orange Micro	
Grappler +	$149.88
Practical Peripherals	
16K Microbuffer II	$224.88
32K Microbuffer II	$254.88
Rana Systems	
Elite 1 Disk Drive	$309.88
Elite 2 Disk Drive	$529.88
Disk Controller	$109.88
Saturn Systems	
32K RAM Board	$204.88
64K RAM Board	$359.88
128K RAM Board	$504.88
SSM	
AIO Multi-Function Card	$199.88
Videx	
VideoTerm	$249.88
VideTerm Combo	$269.88
Enhancer II	$119.88
Vista	
A-800 Disk Controller for 8" disks	$509.88
Vision 80 Card	$269.88

Information & Orders
(603)-673-8857

Orders Only: (800)-343-0726

NO HIDDEN CHARGES

FREE UPS shipping on all orders—No extra charge to use credit cards—All equipment shipped factory fresh with manufacturer's warranty—COD orders accepted ($10 fee added)—No purchase orders accepted—No foreign or APO orders accepted—Minimum $50 per order—This ad prepared in November: prices subject to change.

HIGH TECHNOLOGY AT AFFORDABLE PRICES

MasterCard VISA

THE BOTTOM LINE

MILFORD NH 03055-0423

The Word Handler

THE WORD HANDLER™ SILICON VALLEY SYSTEMS, INC.

A versatile word processor for the Franklin/Apple II. It requires no additional hardware to generate upper/lower case or 40 & 66 columns on-screen. Underlining, boldface and superscripts are displayed on the screen exactly as they will print out; microspacing for justification is also displayed when desirble for printers with microspacing capability. Disk operations are handled automatically, without your intervention (except for disk I/Os). Editing text is easy & fast, with features that include delete, copy, move, find, search, replace, etc. A very capable program.

Word Handler $179.88

List Handler

Stores 3000 records, including letters, lists, labels, etc. Unlimited sort fields and multiple disk drives allows 24,000 records to be handled on-line. DIF format. Stands alone or interfaces to **Word Handler** program.

List Handler $79.88

Quic-N-Easi Applications Generator



The new **Quic-N-Easi AG** allows non-programmers or the novice to write applications for the Franklin/Apple II. If you can define your applications, you can use the **Quic-N-Easi AG.**

The method uses a "fill in the blanks" approach. You create fields, records, & files without using complex databases. Applications are generated to accept input, data entry, edit, update, etc. Information from a database can be manipulated by you using **Quic-N-Easi AG**. The Report Generator formats on-screen, allowing you horizontal, vertical scroll and complete control of on-screen formats for checks, mail labels, invoices—any size, any format.

The **Quic-N-Easi AG** comes complete with AG Program diskette, Report Generator, a **Quic-N-Easi AG** manual & reference card, and sample program lessons. A 64K Franklin/Apple II, with two drives and an 80 column card, are required.

Quic-N-Easi AG (Z80) **................ $210.88**

APPLIED SOFTWARE TECHNOLOGY
Versa Form $298.88
Versa Form/HD $377.88

ARTSCI
Magic Window $84.88
Magic Mailer $62.88
Magic Words $62.88
Magic Pkg. Combo $177.88
Magic Window II $121.88

ASHTON-TATE
dBase II (Z80) $529.88

BRODERBUND
Payroll $303.88
Receivables $377.88

COMMSOFT
Roots/M (Z80) $154.88

COMSHARE
Target Planner Calc $47.88
Master Planner Calc $251.88
Applications Package $47.88
Planner Calc Package $77.88

CONTINENTAL SOFTWARE
Home Accountant $66.88
1st Class Mail $66.88
CPA #1-GL $195.88
CPA #2-AR $195.88
CPA #3-AP $195.88
CPA #4-Payroll $195.88
CPA #5-Proprty Mngmnt $377.88

DAKIN 5
Programming Aid $77.88
Depreciation Planner $303.88
Budget Planner $121.88
Business Bookkeeping $303.88

DATAMOST
Tax Beater $106.88
R.E.A.P. $106.88
Write-On $106.88
Personal Ledger $106.88
Market Technician $106.88
Property Manager $229.88

DATASOFT
Datasm 65 2.0 .... $77.88
LISP Interpreter $103.88

DENVER SOFTWARE
Pascal Tutor $106.88
Pascal Programmer $229.88
Financial Partner $195.88

HAYDEN SOFTWARE
Pie Writer $121.88
Pie Writer/Double $121.88
Pie Writer/Multi-80 .... $121.88
Applesoft Compiler Plus $84.88

DC HAYES
Terminal Program $84.88

HOWARD SOFTWARE
Creative Financing $155.88
NY/NJ State Tax $66.88
IL State Tax $66.88
CA State Tax $66.88
Tax Preparer 82 .... $121.88
Real Estate Analysis $155.88

INSOFT
TransForth II $103.88
ALD System II $66.88
GraForth II $66.88

INSOFT
MatheMagic $77.88
GraphMagic $77.88
Prism $29.88
Mathe/Graph Magic $121.88

IUS
Datadex $121.88
Pro EasyWriter $140.88
TellStar II $69.88
FORTH Development $114.88
Pro EasyMailer $121.88
EasyMover $84.88
Datadex/HD $232.88
Pro EasyWriter/ EasyMailer $251.88

KENSINGTON
Format II $195.88

LINK SYSTEMS
Dataflex $158.88
Datalink $84.88
Link Index $121.88

LJK
Letter Perfect $121.88
Data Perfect $84.88
Edit 6502 .... $84.88

LOTUS
Executive Briefing $158.88

MICRO LAB
Invoice Factory $158.88
Merger $47.88
Learning System $121.88
VisiBlend $47.88
Visifactory $66.88
Asset Manager $158.88
Tax Manager $121.88
Data Factory $232.88

MICRO LAB (cont.)
Wall Streeter $232.88
Payroll Manager $232.88

MICROPRO
WordStar (Z80) $377.88
DataStar (Z80) $168.88
MailMerge (Z80) $76.88
SuperSort (Z80) $115.88
SpellStar (Z80) $115.88
CalcStar (Z80) $115.88
WS Custom zation $273.88
Word/Mail/Spell $636.88
WordStar/MailMerge $488.88
Word/Spell Star $488.88

MICROSOFT
TASC Compiler $140.88
FORTRAN-80 (Z80) $155.88
A.L.D.S. Z80) $103.88
BASIC-80 Compiler (Z80) $303.88
COBOL-80 (Z80) $565.88
muLisp/muStar-80 .... $158.88
M-Sort 80 (Z80) $155.88
muMath/muSimp-80 .... $195.88
ADIOS muMath/muSimp-8 .... $195.88
Multi-Plan (Z80) $214.88
Multi-Plan (Apple DOS) $214.88

MUSE
Data Plot $54.88
Address Book $47.88
Form Letter $84.88
Super Text 40/80 .... $140.88
Super Text 40/56 .... $103.88

ON-LINE
General Manager $181.88

PENGUIN SOFTWARE
Complete Graphics Pkg. $62.88
Special Effects $40.88
Graphics Magician $55.88
Fonts & Characters $24.88

PEACHTREE
PeachPak 40 (Z80) $414.88
40-Inventory (Z80) $282.88
40-Payables (Z80) $282.88
Peachtext (Z80) $350.88
ProofReader (Z80) $214.88
PeachCalc (Z80) $265.88
Mailing List (Z80) $265.88
Telecommunications(Z80) $265.88
PeachPak 80 (Z80) $414.88

PEACHTREE (cont.)
80-Inventory (Z80) $282.88
80-Payables (Z80) $282.88
Text/Spell/Mail (Z80) $214.88

QUALITY SOFTWARE
GBS (Z80) $492.88

SELECT
Select WP $377.88

SOF/SYS
Exec. Secretary $195.88
Exec. Speller $66.88

SOFTECH
UCSD P-system $473.88
BASIC & RunTime $177.88
Xenofile $47.88
Softeach $103.88

SOFTWARE PUBLISHING
PFS $103.88
PFS: Report $81.88
PFS: Graph $103.88

SORCIM
SuperCalc (Z80) $229.88
SpellGuard (Z80) $155.88

SOUTHEASTERN
Data Capture $58.88
Data Capture 80 .... $77.88

STC
Pro Time & Billing $303.88
Reading Development $140.88
Apartment Manager $251.88
Inventory $214.88
IFO $158.88

STONEWARE
D.B. Master $180.88
DB Pak #1 .... $84.88
DB Master/HD $380.88
DB Pak #2 .... $84.88
DB Stat Pac $84.88

SUBLOGIC
Graphics Combo $99.88

SYSTEMS PLUS
General Ledger $288.88
GL/AP/AR $627.88
GL/AP/AR/ & Inventory $782.88

TAURUS
CP Plus $84.88

VISICORP
Desktop Plan $211.88
VisiTrend/Plot $251.88
VisiDex $211.88
VisiTerm $90.88
VisiCalc 3.3 .... $208.88
VisiFile $211.88
VisiSchedule $251.88
Business Forcasting Model $90.88
VisiLink $211.88

Information & Orders
(603)-673-8857
Orders Only: (800)-343-0726

NO HIDDEN CHARGES

FREE UPS shipping on all orders—No extra charge to use credit cards—All software shipped factory fresh & sold as is—No software returns acceptedt—COD orders accepted (please add $10 fee)—No purchase orders accepted—No foreign or APO orders accepted—Minimum $50 per order—Prices subject to change.

HIGH TECHNOLOGY AT AFFORDABLE PRICES

MasterCard VISA

THE BOTTOM LINE

MILFORD NH 03055-0423



BECAUSE READ ONLY MEMORY WILL NEVER FORGET A FIRMWARE PROGRAM, IT WILL STILL BE THERE WHEN THE APPLE IS TURNED OFF AND ON AGAIN.

'FID' ROM

(WITH VOLUME NUMBER SELECTION FOR HARD DISK AND FLOPPY DISK USERS)
THERE IS TWO WAYS TO MOVE TEXT FILES, BINARY FILES, APPLESOFT FILES, INTEGER FILES AND 'R' FILES, FROM ONE DISK TO ANOTHER. THE HARD WAY AND THE EASY WAY!

The easy but inconvenient way is with the 'FID' program supplied by Apple Computer on the DOS 3.3 system master disk. Now a third way is available which is the easy and convenient way - a resident 'FID' in ROM (Read Only Memory). This means you don't have to load it into memory. When ever you need it, it's there! The non essential functions have been removed from the original program leaving the three primary requirements; FILE TRANSFER (any type), CHANGE SOURCE and DESTINATION locations, and CATALOGING of source and destination disks. If the original 'FID' software program just didn't do the trick for you hard disk users then you will appreciate the 'FID' ROM. Volume number selection has been incorporated to allow file transfer between volumes and will also allow file transfers between hard disks and floppy disks and vice versa, regardless of disk controllers being in different slots. The wild card feature for file names is fully supported along with all error trapping features including volume mismatch .... **Price $39.95**

DISK COPY ROM

A RESIDENT DISK COPY PROGRAM WILL MAKE IT EASIER, MORE CONVENIENT, AND WILL ENCOURAGE YOU TO MAKE BACKUPS ON YOUR VALUABLE DISKS.

DISK COPY allows you to copy a disk with one disk drive, or, two disk drives with one controller, or, two disk drives and two controllers. Disk initialization and the copy volume number are selectable options. Two types of copy options are available, gross copy and disk content copy. The gross copy feature will copy every sector on a disk regardless of what or where data is located. This allows you to duplicate data disks which do not have a catalog directory. The disk content option will copy only those sectors which are in use as indicated in the catalog directors. DISK COPY ROM will report all disk drive errors with option to abort at error detection or continuing. DISK COPY ROM will not duplicate copy protected disks ... **Price $39.95**

COMMAND ROM

A RESIDENT CATALOG DIRECTORY MENU WITH ONE KEY CONTROL FOR PERFORMING ALL COMMANDS OF FILE LOADING, RUNNING, DELETING, LOCKING, UNLOCKING, CHANGING DRIVE NUMBERS, DISPLAYING A TRACK/SECTOR MAP, AND DISPLAYING THE START AND LENGTH DATA FOR A BINARY FILE.

When the COMMAND ROM is activated, the disk in the current drive or volume of a hard disk is read to get its complete directory listing. A file command menu is then displayed along with the following information; current drive number, number of sectors used and left, the number of pages which have been set up to hold all the catalog file names, and the first page of file names is displayed. Each page will display up to 16 file names and each file is assigned a letter for identification and ease of access. The space bar can then be pressed to display each new page of file names until all 105 file names have been displayed - if that many exists in the file directory. The COMMAND ROM will then return to the first page for you to review the complete directory listing again. This is the ideal way to find, control, and perform file commands with one key operation. The COMMAND ROM is easily used from within a greeting or hello program .... **Price $39.95**

ANY ONE OF THE ABOVE ROMS CAN BE PURCHASED AS A SYSTEM. THE SYSTEM INCLUDES THE ROM AND SOFT CTRL SYSTEM'S ROM BOARD. EACH SYSTEM PRICE IS.......... $59.95
SOFT CTRL SYSTEM'S ROM BOARD WITHOUT A ROM.......... $29.95

All of the above ROMS will not support 80 column boards and can not be used with relocated versions of DOS.

(Check or M.O.) Visa or MasterCard Accepted

PLEASE ADD $3.00 FOR SHIPPING

SOFT **CTRL** SYSTEMS, BOX 599, WEST MILFORD, NJ 07480 **201-728-8750**

*REGISTERED TRADEMARK
ALL FIRMWARE IS COPYRIGHTED

Circle 18 on Reader Service card.

Saga of Super-Text 40/80

"With Super-Text 40/80, the reviewer explains that the proof is in the using."

by Leslie R. Schmeltz

Muse Software recently introduced an updated version of their popular Super-Text word processing program for the Apple II. Since version 1.0 was introduced some time ago, I have used Super-Text for virtually all of my word processing needs. Each successive revision has added features, refined basic operations and generally kept up with the state of the art. This latest release is no exception. Although its primary selling point is the 80-column display capability, Super-Text 40/80 offers a host of interesting features.

System Requirements

Super-Text 40/80 requires an Apple II or Apple II Plus with 48K RAM, Applesoft ROM, at least one Disk II, and either a lowercase adapter or Videx 80-column board. Muse offers lowercase adapters for those who don't already have one. The shift key modification is also supported, with both the wire and necessary instructions included in the Super-Text 40/80 package.

Documentation

When Super-Text II was released, one of the biggest improvements over the original program was the documentation. Super-Text 40/80 has taken that a step further with even more concise explanations, clear examples and meaningful summaries. Though not set up as a tutorial, the manual provides basic information that, together with Instructions files on the program disk, allows the beginner to get started in using the program.

Interfacing

I am delighted to report that Super-Text 40/80 and my Apple hit it off beautifully—no interfacing problems. In using the two previous versions of Super-Text, I did have some difficulty getting the program to work with my printer and parallel interface card. Granted, Muse solved these minor problems quickly—but the absence of difficulties in Super-Text 40/80 is refreshing.

Muse has gone a long way in simplifying the process of interfacing the program to your computer and peripherals. After selecting the Options program from the title page, you can alter any of four groups of ^Params file values: Printer Interface Parameters, Default Page Format, Default Page Numbering and Special Control Sequences.

The first three groups are self-explanatory. The Special Control Sequences control the special features of your printer. Six control-character sequences may be defined as either printing or nonprinting control codes. Following the on-screen prompts simplifies this process.

In addition, you may select a predefined ^Params file for many of the more popular printers: Epson MX-80, Epson MX-80 with Graftrax, Diablo compatible, Centronics parallel (using standard Apple card), NEC and IDS. Control key sequences for special features contained in these printers are already defined, but may be easily altered by the user.

Cursor Mode

Cursor mode is the "idling state" of the program and controls access to all of the other features. Disk operations are quite logical—press control-L to load a disk file and control-S to save one. Files may be selected by either name or catalog number. The disk catalog, in addition to showing the file names and numbers, also shows the amount of free space on the disk, the name of the last file accessed, the operation selected and slot, drive and volume of the disk. When the disk catalog is displayed, you may also select a different slot and drive, or execute DOS commands Delete, Lock, Unlock and Rename.

The Cursor mode includes commands for cursor and text movement. Delete commands retain the ability to delete single characters, words, lines,

Address correspondence to Leslie R. Schmeltz, 3224 Magnolia Court, Bettendorf, IA 52722.

video "pages" and all information in memory. Several of the delete commands have been simplified and made more logical (control-W to delete a word, for instance).

The Find and Replace operations have always been one of Super-Text's strengths. These abilities have been enhanced by the addition of a wild-card character (!) and the ampersand (&) function to ignore spacing. Multiple words or phrases may be found and replaced in one operation; fully automatic or individually approved replacement is offered. When the process is completed, the program tells you how many occurrences were replaced. The counting feature also counts the occurrences of a specific word or phrase in a text file, and provides an approximate total word count for the file.

Split-screen editing allows you to edit one portion of a file while you examine another. The portion of the screen containing the cursor is active, while movement between screens is a matter of pressing B or T for bottom or top screen respectively.

Block operations allow the user to mark a portion of text for manipulation in some special way. Previous versions of Super-Text have allowed you to copy, save, delete or unmark blocks of text. Super-Text 40/80 has added a block move feature which inserts the block at the cursor and simultaneously deletes it from the original location. Blocks may be saved in separate files on the disk or appended to any file.

As in previous versions, Super-Text 40/80 includes a System Query provision. Pressing Q displays the status of Autolink, the status of The Key, number of occurrences replaced in the last Replace operation and the Super-Text module currently in memory. While the status line is displayed, the user may initialize a data disk by pressing control-I, toggle the Autolink on and off, toggle The Key on and off, define The Key or switch to preview, form feed, or sheet feed print modes. After exiting the System Query, a file status line shows the number of characters currently in the file, the name of the file in memory, and the number of characters you can add to the file.

A couple of the features mentioned in the System Query may be unclear to you. Autolink is a method of linking files together for multiple file Find, Replace and printing operations. Short statements called "links" are inserted at the end of each file of a desired multiple set. With Autolink ON, multiple files from multiple drives may be combined as one large text file.

The Key is a designation for a shorthand provision included in earlier versions of Super-Text. With The Key toggled on, each press of the colon would type "the" on the screen. This feature has been expanded to allow the user to define any sequence of up to 30

characters to be printed when the colon is pressed.

Add Mode

Pressing control-A selects the Add mode, which creates a new text file or adds to an existing one. The word "Add" appears in the lower right corner of the display and, in 40-column mode, the cursor changes to a flashing underline. In this mode, the shift keys (provided you have modified your Apple), The Key, tabs, paragraph indentation and other special features are operational. A type-ahead buffer ensures that you will no longer type faster than the program can accept characters.

If you select Add mode with text beyond the cursor, a series of dashes is inserted and the existing text moved down. As characters are typed, they replace the dashes. Exiting Add mode removes the extra dashes and closes the gap between the new and existing text. If no text is found beyond the cursor, the dashed lines do not appear. Pressing the escape key twice returns you to the Cursor mode.

Change Mode

Perhaps the least glamorous of the various modes, Change simply replaces the character at the cursor with one typed from the keyboard. An echo verifies that replacement has taken place. Cursor and text movement commands are available during the changing process. This mode is indicated by the word "Change" at the lower right of the screen, and is exited by pressing escape twice.



Math Mode

To use this mode, the Math module must first be loaded by pressing control-L and selecting the @Math file. Once this has been done, pressing E or control-E selects the Math mode. Calculations on numbers contained in a file, or those entered directly from the keyboard, may be done in Math mode. Columnar addition and alignment capabilities are included.

Print Mode

When the system is booted, Print mode is resident in the system. Unless Math mode has been used, further disk operations are unnecessary before pressing X to enter this mode. If Math has been used, the @Print file must be loaded from disk.

Default parameters control print format in the system or a format line in a text file. The format line may contain information on left and right margins, paragraph indentation, page length, page numbering, spacing, justification and page number locations. Any of the parameters may be adjusted in the body of the text.

Tabs, temporary margin resets, page breaks and special numbering provisions are included. New in Super-Text 40/80 are provisions for printing page headers and footers. A system of specifications allows you to specify top margin headers, bottom margin headers, alternating headers, and so on.

Form feed or single sheet printing are user selectable. Provisions are included for interrupting and terminating printing operations if problems are encountered. Pressing X to enter Print mode will display the Number of Copies? prompt. You may type the number desired, press 0 (return) to print nothing, or press return and then select multiple files to be printed from the disk catalog that is presented.

The Print Preview mode added to Super-Text II has been expanded in Super-Text 40/80 to include a full 80-column display using the Videx board. All of the operations are the same in either 40- or 80-column mode: the only difference is the displayed line width.

Special features defined for your printer are supported in Print mode, although many of them may not be directly displayed in the Preview mode. Variable character sizes, superscripting, subscripting, emphasized printing and other special printer capabilities do indeed work as advertised.

Impressions

Super-Text 40/80 has carefully retained all the good points of earlier versions of Super-Text and has diligently attempted to eliminate major shortcomings. I particularly like the new multiple Find and Replace, Page Headers and Footers, expansion of The Key, and improved printer interface routines. The addition of 80-column display capabilities puts Super-Text 40/80 in the big leagues. Not requiring an 80-column card shows some concern for the user who may be unwilling, at least initially, to invest the extra dollars for that capability.

Muse seems to have taken a close look at some past negative comments and tried to eliminate as many of these concerns as possible.

Super-Text weaknesses mentioned in earlier reviews were a lack of print preview, 40-character display limitation, no provision for printing headers and footers, no way to insert text from the keyboard during printing, Math mode (rather than Print) resident when booting the system, and no way to display the default format on demand. Additional problems that faced Super-Text II were lack of vertical centering and tabbing. Now, Super-Text 40/80 has trimmed this list still further. No provision for displaying the default format during text entry and no vertical centering remain the only drawbacks.

At the outset of this review I identified myself as a frequent user of all the various versions of Super-Text. The transition from Super-Text II to Super-Text 40/80 was relatively painless—particularly since all the changes were neatly outlined in the front of the manual.

The Muse Form Letter Module provides personalized form letter and document capabilities for Super-Text 40/80.

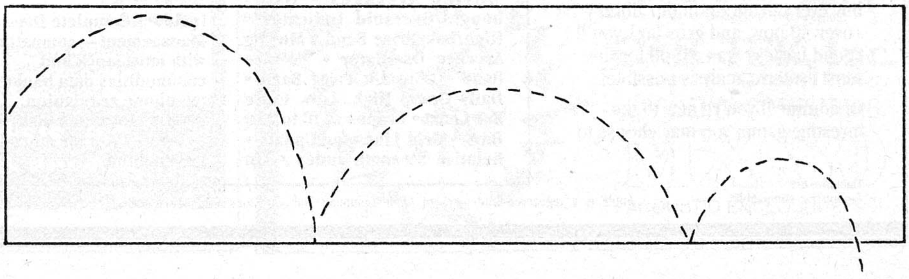
Muse still includes two copies of the program disk and has expanded its replacement policy. Any damaged disk will now be replaced, with no time limit, for $10.

Super-Text 40/80 is one of those programs that rewards effort with professional results. Any good word processing program should be invisible to the recipient of the end product—Super-Text 40/80 is just that.

The latest edition of Super-Text makes an already good program even better. If you are serious about word processing, I encourage you to investigate this program further.

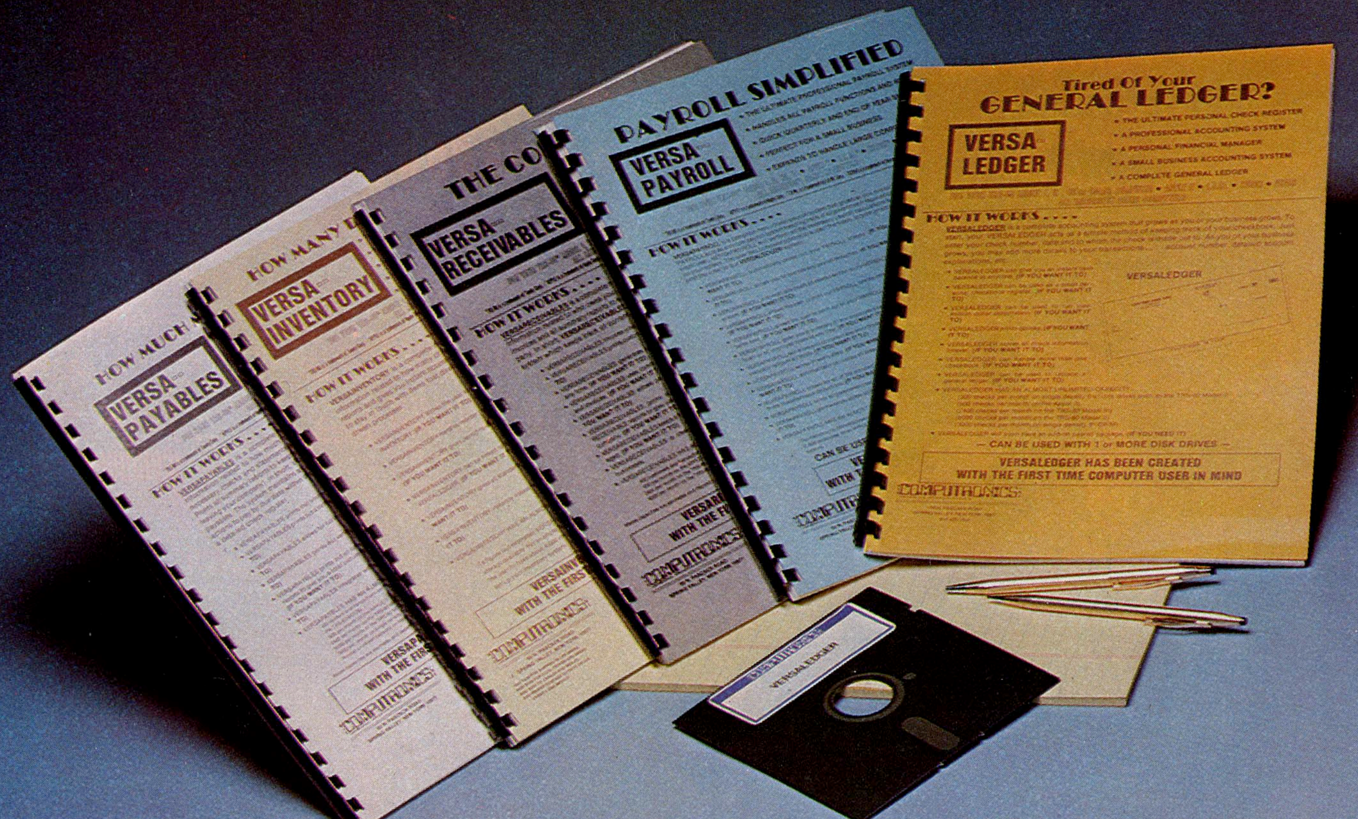
Super-Text 40/80 sells for $175 and is published by Muse Software Inc., 347 N. Charles St., Baltimore, MD 21201. ■

Leslie R. Schmeltz
Bettendorf, IA



Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • XEROX™



The VersaBusiness™ Series

Each **VersaBusiness** module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VersaReceivables™ $99.95

VersaReceivables™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VersaReceivables™ prints all necessary statements, invoices, and summary reports and can be linked with VersaLedger II™ and VersaInventory™.

VersaPayables™ $99.95

VersaPayables™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VersaPayables™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VersaPayables™, you can even let your computer automatically select which vouchers are to be paid.

VersaPayroll™ $99.95

VersaPayroll™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VersaLedger II™ system.

VersaInventory™ $99.95

VersaInventory™ is a complete inventory control system that gives you instant access to data on any item. VersaInventory™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VersaReceivables™ system. VersaInventory™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

COMPUTRONICS INC.

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

VersaLedger II™ $149.95

VersaLedger II™ is a complete accounting system that grows as your business grows. VersaLedger II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VersaLedger II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VersaLedger II™ comes with a professionally-written 160 page manual designed for first-time users. The VersaLedger II™ manual will help you become quickly familiar with VersaLedger II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

Circle 66 on Reader Service card.

SATISFACTION GUARANTEED!

Every **VERSABUSINESS™** module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VERSABUSINESS™** module, you may return it within 30 days for a refund. Manuals for any **VERSABUSINESS™** module may be purchased for $25 each, credited toward a later purchase of that module.

To Order:

Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

VISA · AMERICAN EXPRESS · Master Charge

DEALER INQUIRIES WELCOME

All prices and specifications subject to change / Delivery subject to availability.

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. · *APPLE is a trademark of Apple Corp. · *IBM is a trademark of IBM Corp. · *OSBORNE is a trademark of Osborne Corp. *CP/M is a trademark of Digital Research · *XEROX is a trademark of Xerox Corp.

The Compleat Word Processor

Similar to the Phoenix, which arose from the fires of destruction, ScreenWriter II arises from the ashes of SuperScribe II as a more powerful word processor. The saga of this product's improvements make for interesting reading, for both business and home user.

by G. R. Brieger

ScreenWriter II (previously SuperScribe) is a character and disk oriented word processor that tries to be all things to all writers. Amazingly, it just about succeeds.

To start with, no extra hardware is required to display upper/lowercase characters. An inexpensive modification in the motherboard makes the shift key a real shift key. You can also display up to 70 characters on one line. Unfortunately, this does not really display in the editor mode what the final printout will look like. You have to use the preview function in Runoff to do that. More about that later. The text display is accomplished by two sets of hires characters: 40/line or 70/line.

ScreenWriter II is a perfect example of a good-news/bad-news product. Fortunately, the software's good points far outweigh its bad ones.

For instance, this word processor has more features for the Apple than most do, this side of WordStar, including several even WordStar does not have. All for a cost of only $130. But don't expect to quickly master the whole program. It's so large that even the table of contents is divided into four sections.

The revised documentation consists of 187 pages of text, including an alphabetical index. Section I of the table of contents, The Mini-Writer, is written for the beginner. A tutorial on the disk plus the Mini-Writer will get you started.

The commands are almost always mnemonic. Nice, but you have to have an eidetic memory to recall the appropriate command at the right time. There are over 150 commands, of which the majority are for designing highly specialized formats. So, you look them up on one of three special Command Cards. Due to the large number of choices, the program quickly runs out of available control characters. Instead, you use a period plus two letters, followed by a carriage return, for a total of four keystrokes. This makes for cumbersome operation.

I opened by mentioning that ScreenWriter II is character and disk oriented. A character oriented word processor is quick to respond, especially when you are typing manuscripts with extensive and repeated changes. ScreenWriter II excels in its editing features. Also, a file may contain as many as 65,000 characters. The program accomplishes this by writing text to disk frequently. Unless you watch for this operation, it's transparent to the typist. This was a problem with earlier versions of SuperScribe, now seemingly solved. I never lost any text when composing a lengthy document.

A warning in the manual cautions never to remove a data disk while editing is in progress. Heed this advice. Past problems in losing text may simply have been caused by disk swapping.

The disk drive, or drives, are active more often than I am used to. A second drive certainly comes in handy, but you can manage quite well with only one. If you have two disk drives, customizing the program disk will automatically activate the correct drive for loading and saving files.

A type-ahead buffer makes losing characters, even for the fastest typist, nearly impossible. Should you fill the buffer, the computer will beep at you.

Formatting Magic

Any imaginable type of formatting can be accomplished. In addition to the usual setting of right, left, top and bottom margins, and all kinds of tabs, you also have right justification, centering and proportional spacing. Need sub- and superscript? Or how about changing ribbon color from black to red and back again? No problem. If you have an Epson MX-70, 80 or 100, with or without Graftrax, you can change fonts on the fly. The manual is not specific on how this is done.

The control codes for the Epson MX-80 printer are shown in the table. Some will work only if you have the

Address correspondence to Gerry Brieger, 2716 174th Ave. NE, Redmond, WA 98052.

Table 1.
Control codes for Epson MX-80 printer.

<ESC>	P	compressed mode	ON
	Q	compressed mode	OFF
	4	italics mode	ON
	5	italics mode	OFF
	E	emphasized mode	ON
	F	emphasized mode	OFF
	G	bold mode	ON
	H	bold mode	OFF
	S	expanded mode	ON
	T	expanded mode	OFF

graphics package. You can combine appropriate modes. For instance, the compressed italic characters are impressive as footnotes.

To change characters on the fly, type control-X, then escape. The control-X is invisible. ScreenWriter II uses a bell symbol for escape; simply follow it with the letter from the above table. At the end of a word or phrase, the same procedure is repeated.

The bell and the letter will not be printed, but are recognized by the printer to produce the font you want. If you want your text justified, there's a slight problem, as the program counts control characters as spaces.

ScreenWriter II also has many sophisticated capabilities. Would you like page numbering in Roman numerals? It's possible anywhere on the page outside the normal text area. Footnotes, page headers and dates are printed where you want them. There's also a system for form letters and various types of hyphenation. Don't try to hyphenate long documents in manual mode, unless you're willing to spend a lot of time watching prompts while the file is printed.

Bonus Points

If you're writing a lengthy document, or perhaps a book, ScreenWriter II will produce several types of indexes. If you want to know how many characters or words you have typed, just ask the program. Spooling (editing one file while another one is printing) is provided for some printers. The modular system of the ScreenWriter II program requires that you load Runoff when you want to print a file. This is time consuming, especially when learning this program. The time required to move between Editor and Runoff is almost too long. . . unless you have the language or 16K RAM card in slot #0. If you have the extra RAM, Runoff is loaded there, and the extended waiting time between editing and printing is eliminated. I would not care to use ScreenWriter II without this extra memory.

Macros are easily obtained with a few keystrokes at the start of editing. Since I was going to type "ScreenWriter II" many times, I created a macro, as shown in the manual, by substituting a control-0. It worked perfectly the first time. I am also familiar with having a symbol that isn't used often print "the" with one keystroke. I selected the semicolon for this purpose, which speeded up my typing considerably. Macros are convenient.

It's also easy to transmit ScreenWriter II files over the telephone. Electronic Mail for the Apple is here. You can print a formatted version of a text file to disk, as well. All is explained in the manual, but to me seemed a bit complicated.

In the appendix you are instructed on how to convert SuperText II or Applewriter files to ScreenWriter II text files. This can be handy in case you want to change word processors.

ScreenWriter II requires an Apple II or II+, 48K of RAM, DOS 3.3 and one disk drive. An extra 16K of memory, a second disk drive and the shift key modification make life a lot easier. Multiple drives are supported.

To sum it all up, ScreenWriter II does a great deal, but don't expect to use all of its features unless you're willing to spend some time learning how. If you make the commitment, you might never need another word processor. ■

Statistics for the Masses

Our author shows how statistics can be managed easily and painlessly, and rewards the reader with a program that will generate these meaningful numbers.

by Steven A. Schwartz

Regardless of background or profession, most people have a fairly good idea about what the term "average" implies. Statistically, it is calculated as the sum of all cases divided by the number of cases. When you take an average of a set of numbers, you are deriving a single statistic which summarizes those numbers. Summary statistics such as this are useful in communicating to others what a set of data looks like without having to relate each individual item.

For example, it is simpler to say that the average monthly electric bill for the last year was $43.65 than to list all 12 bills separately. This becomes even more apparent (and essential) when you work with a large number of cases (e.g., the average height of American men).

Although such statistics are most frequently encountered in news articles and research or survey results, they are equally useful in business settings (to describe sales of an item over a given period of time, to summarize cash receipts, etc.) and in the home (utility bills, budget calculations, etc.).

Our society is exceptionally number-oriented. We look to the Dow-Jones Industrial Average to determine whether stocks rose or fell for the day. We watch the weather on local television to see how today's temperature compared with the typical temperature for this time of year, and what the average rainfall has been. We examine the Consumer Price Index to see how our standard of living has fared and what to expect in the near future.

The following program for summary or descriptive statistics is intended to help you generate some meaningful numbers of your own. As you examine the program and run it with your own data, you'll develop an intuitive grasp of the meaning of the statistics and a "feel" for how they relate to one another.

System Requirements and Program Output

I wrote the program shown in the listing in Applesoft Basic for a 48K Apple II Plus with a single disk drive and 132-column printer. The disk is necessary only if you wish to save your data set(s) for further analysis (e.g., adding more cases as they become available). With a little work, it should be relatively easy to convert the program for use with a cassette-only system. I'll provide suggestions later in the article for use with an 80-column printer—or no printer at all.

Data Description 2.0 (see program listing) calculates summary statistics for a maximum of 20 variables and 225 cases at a time. It provides titled and dated output, automatic page numbering, variable labels, reviewing/editing of the data (either all cases or selected cases only), saving data to and reading from disk, and error handling.

Statistics for each variable include the sum of all cases, the mean (arithmetic average), the maximum and minimum values, the range of cases, the variance, standard deviation, standard error of the mean, skewness, kurtosis, and the number of cases. In addition, the program also performs a frequency count for each variable and prints a histogram (bar graph) to show you what your data distribution looks like.

Explanation of the Statistics

To best understand the statistics available with Data Description 2.0, look at this short example. Suppose you wanted to see how many cigarettes you smoked in a two-month period. To do so, you would enter the number smoked for each of 60 days. After running the program, the results might look like those in Figure 1.

The sum (2002) is the total number of cigarettes smoked during the 60 days. The mean (33.366667) is the average number of cigarettes smoked per day over the 60-day period. Note that the mean is calculated by dividing the sum by the number of cases (2002/60).

Address correspondence to Steven A. Schwartz, Ph.D., 9226 Vantine St., Pittsburgh, PA 15235.

Figure 1.
Sample output of Data Description program.

```
********
* CIGS *
********

SUM      = 2002                        VARIANCE     = 49.761582
MEAN     = 33.3666667                  STAND. DEV.  = 7.05418897
MAXIMUM  = 55                          STAND. ERR.  = .910691879
MINIMUM  = 18                          SKEWNESS     = -.0410452523
RANGE    = 37                          KURTOSIS     = .462266477

NO. OF CASES = 60

VALUE    COUNT    %      CUM. %   HISTOGRAM OF FREQUENCIES
-----    -----    -      ------   ------------------------

18         1     1.67     1.67    * (1)
19         3     5.00     6.67    *** (3)
22         2     3.33    10.00    ** (2)
24         1     1.67    11.67    * (1)
26         1     1.67    13.34    * (1)
27         1     1.67    15.01    * (1)
28         1     1.67    16.68    * (1)
29         9    15.00    31.68    ********* (9)
30         4     6.67    38.35    **** (4)
31         2     3.33    41.68    ** (2)
32         1     1.67    43.35    * (1)
35         8    13.33    56.68    ******** (8)
36         8    13.33    70.01    ******** (8)
37         1     1.67    71.68    * (1)
38         5     8.33    80.01    ***** (5)
39         3     5.00    85.01    *** (3)
40         3     5.00    90.01    *** (3)
42         1     1.67    91.68    * (1)
43         3     5.00    96.68    *** (3)
45         1     1.67    98.35    * (1)
55         1     1.67   100.00    * (1)

-----------------------------------------------------------------
```

The program next prints the maximum (55) and minimum (18) for the variable. Thus, the most cigarettes smoked in a day was 55 and the least was 18. The range (37) is simply the difference between the maximum and minimum (i.e., 55 – 18).

The variance and standard deviation are two statistics that describe the dispersion, or spread, of data values about the mean. The smaller the figures, the closer all data values group around the mean—that is, the more homogeneous the data set. Mathematically, the standard deviation is simply the square root of the variance.

According to the authors of the SPSS (see Reference), if you were to draw an infinite number of equal-sized samples from a given population, the mean of each sample would be an estimate of the true population mean, but not all of them would be identical. The pattern of these means would constitute a normal distribution and would have a standard deviation. The standard deviation of this distribution is the standard error. Thus, the standard error helps us determine the potential degree of discrepancy between the sample mean and the (usually) unknown population mean. In general, we deal only in samples and sample statistics. Our cigarette study, for example, represents only a single sample of all the 60-day periods we might have examined.

Skewness and kurtosis tell us about the shape of the data distribution. A positive value of skewness indicates that the cases are grouped to the left of the mean (low end) with the extreme cases to the right. A negative value indicates the opposite. You'll see a value close to zero (as in the cigarette example) when the distribution is a normal bell curve.

Kurtosis indicates the relative flatness or peakedness of the data curve. A normal distribution will have a kurtosis of zero. If the kurtosis is positive, then the distribution is more peaked (narrow) than would be true for a normal distribution, while a negative value means that it is flatter. The positive value in our example (.46) suggests that the distribution has a slightly higher peak than in a normal curve. To see the shape of the distribution in our example, turn Figure 1 on its side with the bar graph facing up. The two peaks (9 days of 29 cigarettes per day and 8 days of 35 or 36 cigarettes per day) account for the positive kurtosis.

Finally, let's look a little closer at the frequency count and histogram sections of output. In the column labeled VALUE are listed all the values that the data elements have taken for that specific variable. COUNT represents the number of times that the data value to the left appeared in the data set. For example, there were 3 days when 19 cigarettes were smoked, 9 days when 29 cigarettes were smoked, etc.

The % column shows us the percentage of data values that each count represents. For instance, 30 cigarettes were smoked per day on 6.67% of the days of the study. The CUM. % column tells us the cumulative percentage, i.e., the total percentage for that data value plus all other smaller data values. Thus, 41.68% of the time, 31 or fewer cigarettes were smoked per day.

The column HISTOGRAM OF FREQUENCIES is our bar graph corresponding to the obtained data values. It is simply a pictorial representation of the count for each data value, using asterisks. Each asterisk stands for one count of data (up to a maximum count of 85). Any count greater than 85 will be shown by 85 asterisks in a string. Note, however, that the actual count is enclosed in parentheses following the string of asterisks. In most instances, since you are limited to 225 cases per variable, the 85 asterisk limit will not be a problem.

Program Operation

Following presentation of the title page, you will be asked for a label (80 characters or less) for your output—line 110. It's usually best to make your title descriptive of your study or data set—for example, CASH RECEIPTS FOR THE PERIOD OF 04/01/82–06/30/82. Six months later when you look at your printout, you won't have to guess what the analysis covered.

Next, you enter a date for the run. You must use an eight-character string for the date, as in 04/15/82 or 04-15-82.

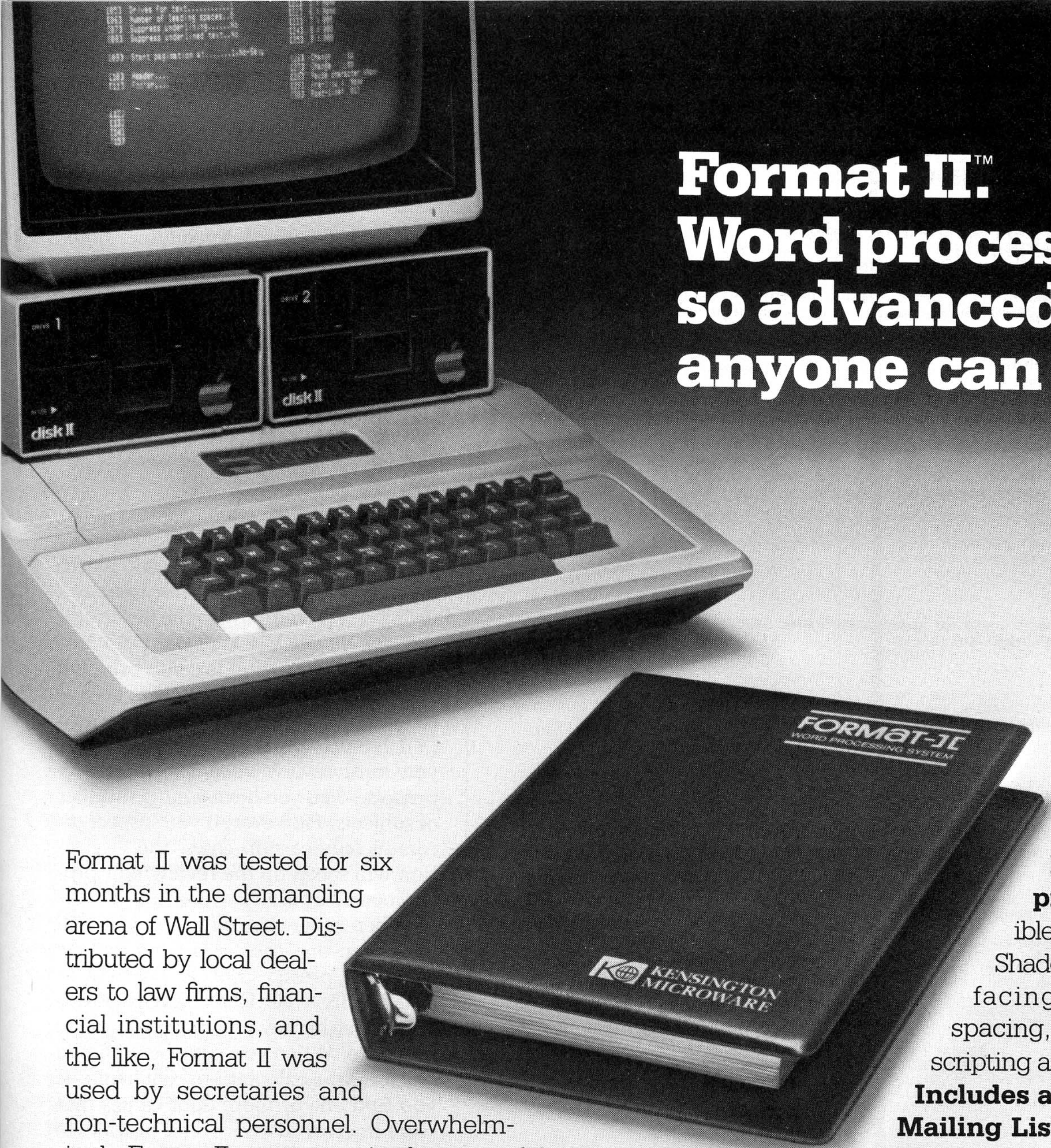
Now you will be asked: ENTER DATA FROM DISK (Y/N)? If you wish to analyze a previously saved data set as is or want to add more data to such a set, press Y. If you're entering all data directly through the keyboard, press N. If you press Y, the program goes to the Disk Read routine beginning at line 1630.

After supplying the name of the data file, the Apple reads the number of variables (NVAR), the number of cases or subjects (N1), the array of variable names (NAME$(I)) and the data set (A(I,J)). If you want to add more cases to the disk data set, answer Y (yes) to the appropriate question and indicate the number of additional cases when prompted. As long as the total number of cases does not exceed the program limit—225—the program will ask you to begin entering the additional data (line 380). If no additional cases were indicated, Data Description will move directly to the edit/review phase (line 460).

If you answered N to the ENTER DATA FROM DISK (Y/N) query, you will be expected to indicate the number of cases, subjects and variable names. Note that the program checks for maximum and minimum allowable values for the first two and a maximum length of six characters for each variable name. Also, if you just press return when asked for each variable name, the program assigns a default value composed of the string VAR plus the variable number; e.g., VAR1, VAR2, etc. As indicated above, when data is read from the disk, this information is read at the same time so it need not be reentered.

Keyboard data input is all per-

Format II.™ Word processing so advanced anyone can use it.



Format II was tested for six months in the demanding arena of Wall Street. Distributed by local dealers to law firms, financial institutions, and the like, Format II was used by secretaries and non-technical personnel. Overwhelmingly, Format II was recognized as one of the easiest and most effective word processing systems available *in any form.*

Here's why Format II is unique among word processing programs:

What you see is what you get. Format II performs virtually any editing and formatting function you can imagine and displays on the screen the text exactly as it will print out—paragraphs, underlining, justification, page breaks.

Simple to Use. You edit and format text with single key strokes. "D" for delete, "E" for edit, "I" for insert, "J" for justify, etc. Easy-to-remember commands because they make sense.

It supports all printers compatible with the Apple. Shadow printing, boldfacing, proportional spacing, sub- and superscripting are all available.

Includes a sophisticated Mailing List. Stores and retrieves names and addresses which may be printed on labels or incorporated into documents. Powerful "logic" commands allow you to select only those records which match specified criteria.

And more. Format II is lightning fast and menu driven. Add them all together and it's clear why Format II has generated such enthusiasm. Now, at a cost of $250, you *too* can buy the best.

Format II is available from most local dealers. If not, ask them to contact us, or order directly and receive a 30 day money-back guarantee.

Kensington Microware Ltd., 300 East 54 St., Suite 3L, New York, NY 10022, (212) 486-2802.

KENSINGTON MICROWARE

System Requirements: 48K Apple® II Plus with 80 column card.

Circle 86 on Reader Service card.

Program listing.
Data Description 2.0.

```
10  ONERR  GOTO 2660
20  REM **************
30  REM * TITLE PAGE *
40  REM **************
50 D$ =  CHR$ (4):N1 = 0
60  DIM NAME$(20),A(225,20),CT(225),VL(225)
70  TEXT : HOME : FOR I = 1 TO 20: PRINT "*=";: NEXT I: PRINT
80  PRINT "D A T A   D E S C R I P T I O N   2 . 0": PRINT : FOR I = 1 TO 20: PRINT "*=";: NEXT I
90  VTAB 23: INVERSE : PRINT "* (C) STEVEN A. SCHWARTZ, PH.D. 1982 *": NORMAL
100  FOR I = 1 TO 3000: NEXT I: HOME
110  VTAB 1: HTAB 1: CALL  - 958: PRINT "ENTER TITLE (2 LINES MAXIMUM):": PRINT : INPUT "";TITLE$
120  IF  LEN (TITLE$) > 80 THEN 110
130  VTAB 6: HTAB 1: CALL  - 958: INPUT "DATE (E.G.. 03/04/82): ";DT$: IF  LEN (DT$) <  > 8 THEN 130
140  VTAB 8: HTAB 1: CALL  - 868: PRINT "ENTER DATA FROM DISK (Y/N)? ";: GET Z$: PRINT Z$: IF Z$ <  > "Y" AND Z$ <  > "N" THEN
     140
150  IF Z$ = "Y" THEN 1630
160  REM ************************************
170  REM * SET NO. OF VARIABLES AND CASES *
180  REM ************************************
190  PRINT : INPUT "ENTER NUMBER OF VARIABLES: ";NVAR
200  IF NVAR < 1 THEN  INVERSE : PRINT "NO. OF VARIABLES MUST BE AT LEAST 1.": NORMAL : GOTO 190
210  IF NVAR > 20 THEN  INVERSE : PRINT "MAXIMUM NUMBER OF VARIABLES = 20": NORMAL : GOTO 190
220  PRINT : INPUT "ENTER NUMBER OF CASES:     ";CASE
230  IF CASE <  = 1 THEN  INVERSE : PRINT "YOU MUST HAVE MORE THAN 1 CASE.": NORMAL : GOTO 220
240  IF CASE > 225 THEN  INVERSE : PRINT "MAXIMUM NUMBER OF CASES EXCEEDED (225)": NORMAL : GOTO 220
250  REM ******************************
260  REM * ASSIGN NAMES TO VARIABLES *
270  REM ******************************
280  HOME : PRINT "VAR. NAMES (RETURN FOR DEFAULT)": PRINT : PRINT
290  FOR I = 1 TO NVAR
300  PRINT  TAB( 5)"VAR. ";I;" = ";
310  INPUT NAME$(I)
320  IF  LEN (NAME$(I)) > 6 THEN  INVERSE : PRINT "MAX. IS 6 CHARACTERS.": NORMAL : GOTO 300
330  IF NAME$(I) = "" THEN NAME$(I) = "VAR" +  STR$ (I)
340  NEXT I
350  REM **************
360  REM * DATA INPUT *
370  REM **************
380  FOR I = N1 + 1 TO CASE: HOME : PRINT  CHR$ (7);
390  PRINT "DATA FOR SUBJECT ";I
400  FOR J = 1 TO NVAR
410  VTAB 4: HTAB 1: CALL  - 958
420  PRINT NAME$(J);" = ";
430  INPUT A(I,J)
440  NEXT J
450  NEXT I
460  VTAB 11: HTAB 6: CALL  - 958: INVERSE : PRINT "EDIT/REVIEW THE DATA (Y/N)? ";: NORMAL : GET Z$: PRINT Z$: IF Z$ <  > "Y" AND
     Z$ <  > "N" THEN 460
470  IF Z$ = "Y" THEN  GOSUB 2030
480  REM ************
490  REM * ANALYSES *
500  REM ************
510  FOR J = 1 TO NVAR
520  PR# 0: HOME : HTAB 9: FLASH : PRINT  CHR$ (7): CHR$ (7)"PERFORMING CALCULATIONS": NORMAL
530 SUM = 0:M2 = 0:M3 = 0:M4 = 0
540  FOR I = 1 TO CASE
550 SUM = SUM + A(I,J):M2 = M2 + A(I,J) ^ 2:M3 = M3 + A(I,J) ^ 3:M4 = M4 + A(I,J) ^ 4
560  NEXT I
570 MEAN = SUM / CASE
580 MAX = A(1,J)
590 Z = 1
600  FOR I = 2 TO CASE
610  IF A(I,J) <  = MAX THEN 640
620 Z = I
630 MAX = A(Z,J)
640  NEXT I
650 MN = A(1,J)
660 Z = 1
670  FOR I = 2 TO CASE
680  IF A(I,J) >  = MN THEN 710
690 Z = I
700 MN = A(Z,J)
710  NEXT I
720 RANGE = MAX - MN
730 SS = 0
740  FOR I = 1 TO CASE
750 SS = SS + (A(I,J) - MEAN) ^ 2
760  NEXT I
770 S2 = SS / (CASE - 1)
780 SD =  SQR (S2)
790 SE = SD / ( SQR (CASE))
800 H = ((M3 - 3 * SUM / CASE * M2 + 3 * (SUM / CASE) ^ 2 * SUM) / CASE) - (SUM / CASE) ^ 3
810 D = ((M2 - CASE * (SUM / CASE) ^ 2) / (CASE - 1)) ^ 1.5:SK = H / D
820 H = ((M4 - 4 * SUM / CASE * M3 + 6 * (SUM / CASE) ^ 2 * M2 - 4 * (SUM / CASE) ^ 3 * SUM) / CASE) + (SUM / CASE) ^ 4
830 D = ((M2 - CASE * (SUM / CASE) ^ 2) / (CASE - 1)) ^ 2:KR = H / D - 3
840  REM ******************
850  REM * HISTOGRAM SORT *
860  REM ******************
870 M = CASE
880 M =  INT (M / 2)
890  IF M = 0 THEN 1060
900 K = CASE - M
910 J1 = 1
920 I1 = J1
930 L1 = I1 + M
940  IF A(I1,J) < A(L1,J) THEN 1000
950 T = A(I1,J)
960 A(I1,J) = A(L1,J)
970 A(L1,J) = T
980 I1 = I1 - M
990  IF I1 >  = 1 THEN 930
```

Listing continued.

formed on the fourth screen line. Both the subject number and variable name are clearly indicated. Whenever data input for a new subject (case) begins, you are prompted by a single beep—CHR$(7) in line 380. This way you can type the data values without constantly checking the screen to see when a new case starts. I find this helpful, but if you don't (or feel that it's annoying), simply rewrite the line as:

```
380 FOR I = N1 + 1 TO CASE: HOME
```

After data entry, if you would like to edit or review the data prior to analysis, press Y in response to the next question (EDIT/REVIEW THE DATA?). This program option lets you examine and/or change all of the data or selected cases. If you press A when asked (A)LL OR (S)ELECTED CASES?, you must review all data—a lengthy process when you have a large number of subjects. However, if you suspect errors in only certain cases, the S selection will speed up the review/edit process considerably.

When reviewing *all* cases, you will be shown one case at a time and be asked to press: <ESC> TO EDIT; <RETURN> TO CONTINUE. Pressing the return key will move you to the next case. When you have been shown all cases, the calculations will begin. If you find one or more data values that were in error or you would just like to change, press the escape key. Then indicate the number of the variable that you wish to change (counting down from the top if you did not use the default variable names; i.e.,VAR1, VAR2, etc.).

Finally, enter the new value for the variable. If you selected the wrong variable, simply reenter its original, correct, value. The screen will now be updated to show the effect of your change. You may continue editing the case, by pressing escape again. Press return when you wish to move to the next case.

The Selected Cases editing mode operates in a similar manner, except that you must indicate the specific CASE # TO REVIEW (OR 0 TO END) prior to editing or reviewing each case. When you press 0, the analysis will begin.

Analyses are performed one variable at a time. When finished for the first

Listing continued.

```
1000 J1 = J1 + 1
1010  IF J1 > K THEN 880
1020  GOTO 920
1030  REM ##################
1040  REM # FREQUENCY COUNT #
1050  REM ##################
1060 CT(1) = 1:VL(1) = A(1,J):L = 1
1070  FOR I = 2 TO CASE: FOR II = 1 TO L
1080  IF A(I,J) = VL(II) THEN CT(II) = CT(II) + 1: GOTO 1110
1090  NEXT II
1100 L = L + 1:CT(L) = 1:VL(L) = A(I,J)
1110  NEXT I
1120  REM ##########
1130  REM # OUTPUT #
1140  REM ##########
1150  IF J = 1 THEN  PR# 0: HOME : FLASH : PRINT  CHR$ (7): CHR$ (7):"TURN ON PRINTER AND PRESS <RETURN>.....";: NORMAL : GET
   Z$: PRINT Z$: IF Z$ < > CHR$ (13) THEN 1150
1160  PR# 1: PRINT  CHR$ (15) +  CHR$ (9) + "132N"
1170  GOSUB 1580
1180  PRINT
1190  PRINT "SUM      = ";SUM;: POKE 36,39: PRINT "VARIANCE     = ";S2
1200  PRINT "MEAN     = ";MEAN;: POKE 36,39: PRINT "STAND. DEV.  = ";SD
1210  PRINT "MAXIMUM  = ";MAX;: POKE 36,39: PRINT "STAND. ERR.  = ";SE
1220  PRINT "MINIMUM  = ";MN;: POKE 36,39: PRINT "SKEWNESS     = ";SK
1230  PRINT "RANGE    = ";RANGE;: POKE 36,39: PRINT "KURTOSIS     = ";KR
1240  PRINT : PRINT "NO. OF CASES = ";CASE
1250  PRINT
1260  REM ##################
1270  REM # PRINT HISTOGRAM #
1280  REM ##################
1290  PRINT "VALUE      COUNT      %       CUM. %";: POKE 36,39: PRINT "HISTOGRAM OF FREQUENCIES"
1300  PRINT "-----      -----      -       ------";: POKE 36,39: PRINT "------------------------": PRINT
1310 CPCT = 0
1320  FOR I = 1 TO L
1330 DOT = CT(I): IF CT(I) > 85 THEN DOT = 85
1340  PRINT VL(I);: POKE 36,13: PRINT CT(I);
1350 PCT = CT(I) / CASE:PCT =  INT (PCT * 10000 + .5) / 100: IF PCT > = 10 AND PCT < 100 THEN  POKE 36,20: GOTO 1390
1360  IF PCT < 1 THEN  POKE 36,22: GOTO 1390
1370  IF PCT = 100 THEN  POKE 36,19: PRINT "100.00";: GOTO 1410
1380  POKE 36,21
1390 X$ =  STR$ (PCT): IF PCT =  INT (PCT) THEN X$ = X$ + ".00"
1400  PRINT X$;
1410 CPCT = CPCT + PCT:CPCT =  INT (CPCT * 100 + .5) / 100: IF I = L THEN CPCT = 100: POKE 36,29: GOTO 1450
1420  IF CPCT < 1 THEN  POKE 36,32: GOTO 1450
1430  IF CPCT < 10 THEN  POKE 36,31
1440  IF CPCT > = 10 AND CPCT < > 100 THEN  POKE 36,30
1450 X$ =  STR$ (CPCT): IF CPCT =  INT (CPCT) THEN X$ = X$ + ".00"
1460  PRINT X$;
1470  POKE 36,39: FOR K1 = 1 TO DOT: PRINT "*";: NEXT K1: PRINT " (";CT(I);")"
1480  NEXT I
1490  PRINT : FOR I = 1 TO 132: PRINT "-";: NEXT I: PRINT : PRINT
1500  NEXT J
1510  PRINT  CHR$ (12): PR# 0
1520  HOME : PRINT "SAVE DATA TO DISK (Y/N)? ";: GET Z$: PRINT Z$: IF Z$ < > "Y" AND Z$ < > "N" THEN 1520
1530  IF Z$ = "N" THEN  END
1540  GOTO 1870
1550  REM ################
1560  REM # PAGE HEADERS #
1570  REM ################
1580  FOR I = 1 TO  LEN (NAME$(J)) + 4: PRINT "*";: NEXT I: POKE 36,131 -  LEN (TITLE$): PRINT TITLE$
1590  PRINT "* ";NAME$(J);" *";: POKE 36,126 -  LEN ( STR$ (J)): PRINT "PAGE ";J
1600  FOR I = 1 TO  LEN (NAME$(J)) + 4: PRINT "*";: NEXT I: POKE 36,123: PRINT DT$
1610  RETURN
1620  REM ##############################
1630  REM # DISK READ (PRIOR DATA SET) #
1640  REM ##############################
1650  ONERR  GOTO 2460
1660  HOME : INVERSE : PRINT "NAME OF FILE:";: NORMAL : INPUT " ";F$
1670  VTAB 3: PRINT "PRESS ANY KEY WHEN YOU ARE READY.......";: GET Z$: VTAB 6: HTAB 15: FLASH : PRINT "<<WORKING>>": NORMAL
1680  PRINT D$;"NOMON C,I,O"
1690  PRINT D$;"OPEN";F$
1700  PRINT D$;"READ";F$
1710  INPUT NVAR: INPUT N1
1720  FOR I = 1 TO NVAR: INPUT NAME$(I): NEXT
1730  FOR I = 1 TO N1: FOR J = 1 TO NVAR
1740  INPUT A(I,J)
1750  NEXT J
1760  NEXT I
1770  PRINT D$;"CLOSE";F$
1780  ONERR  GOTO 2660
1790  HOME : PRINT "ENTER ADDITIONAL DATA FROM KEYBOARD?": PRINT "(PRESS Y OR N): ";: GET Z$: PRINT Z$: IF Z$ < > "Y" AND Z$ <
   > "N" THEN 1790
1800  IF Z$ = "N" THEN CASE = N1: GOTO 460
1810  VTAB 5: HTAB 1: CALL  - 958: INVERSE : PRINT "NUMBER OF ADDITIONAL CASES:";: NORMAL : INPUT " ";N2
1820 CASE = N1 + N2: IF CASE > 225 THEN  INVERSE : PRINT "MAXIMUM NUMBER OF CASES EXCEEDED (225)": NORMAL : FOR I = 1 TO 1500:
   NEXT I: GOTO 1810
1830  GOTO 380
1840  REM ####################
1850  REM # SAVE DATA TO DISK #
1860  REM ####################
1870  ONERR  GOTO 2530
1880  HOME
1890  INVERSE : PRINT "NAME OF FILE:";: NORMAL : INPUT "  ";F$: IF F$ = "" THEN 1890
1900  VTAB 3: PRINT "PRESS ANY KEY WHEN YOU ARE READY.......";: GET Z$: VTAB 6: HTAB 15: FLASH : PRINT "<<WORKING>>": NORMAL
1910  PRINT D$;"OPEN";F$
1920  PRINT D$;"DELETE";F$
1930  PRINT D$;"OPEN";F$
1940  PRINT D$;"WRITE";F$
1950  PRINT NVAR: PRINT CASE
```

Listing continued.

variable, the Apple will beep twice (line 1150) and you will be prompted to turn on your printer and press return. Line 1160 routes output to the printer (PR#1) and sets it to 132-column mode. Note that CHR$(15) is equal to a control-O, the command that sets an Epson printer in condensed print mode. CHR$(9) is a control-I. Make appropriate changes to this line if your printer card is in a slot other than number 1 or you use different control characters to set up 132-column printing.

From this point on, you need not intervene in the process until all analyses are finished. The program will pause after printing the results for each variable. Don't be alarmed; it takes time to make all the necessary calculations—and the more cases, the longer the pauses.

Finally, the program asks if you would like to save the data set to disk. If you do, press Y, and then name the data file when prompted. If you want to write over an old data file, reuse the

''Once you're comfortable. . .''

same filename. Otherwise, pick any name you like, press return, insert a disk, and press return when you're ready to begin the save process.

Error Handling

In addition to screening for an appropriate range of responses when numeric input is required (e.g., lines 200, 210, 230, 240), correct length of character strings (e.g., lines 120, 130, 320), and allowable answers to questions (as in Y for yes or N for no), Data Description 2.0 has three major error routines located between lines 2430 and 2680. The first one, beginning at 2430, screens disk read errors; the second handles disk write errors; and the third is a catchall, handling illegal data input and all other unspecified errors.

The first line of the program (10) sets up the primary Onerr Goto routine and directs all such errors to line 2660. If any error occurs, the printer is shut off (PR#0)—assuming that it was on—and the variable ERR is set equal to the appropriate Applesoft or DOS error code (PEEK(222)).

Listing continued.

```
1960  FOR I = 1 TO NVAR: PRINT NAME$(I): NEXT
1970  FOR I = 1 TO CASE: FOR J = 1 TO NVAR: PRINT A(I.J): NEXT J: NEXT I
1980  PRINT D$;"CLOSE";F$
1990  HOME : END
2000  REM ********************
2010  REM * EDIT/REVIEW DATA *
2020  REM ********************
2030  VTAB 14: HTAB 6: CALL  - 958: INVERSE : PRINT "(A)LL OR (S)ELECTED CASES? ";: NORMAL : GET Z$: PRINT Z$: IF Z$ <  > "A" AND
      Z$ <  > "S" THEN 2030
2040  IF Z$ = "S" THEN 2270
2050  REM *************
2060  REM * ALL CASES *
2070  REM *************
2080  FOR I = 1 TO CASE: HOME
2090  INVERSE : PRINT " CASE #";I;" ": NORMAL : PRINT : PRINT
2100  FOR J = 1 TO NVAR
2110  PRINT NAME$(J): TAB( 8)"= ";A(I,J):
2120  IF NVAR >  = J + 10 THEN  PRINT  TAB( 21)NAME$(J + 10); TAB( 28)"= ";A(I,J + 10): GOTO 2140
2130  PRINT
2140  IF J < 10 THEN  NEXT J
2150 J = NVAR
2160  VTAB 24: HTAB 1: CALL  - 958: INVERSE : PRINT "  <ESC> TO EDIT: <RETURN> TO CONTINUE  ";: GET Z$: NORMAL : IF Z$ <  >  CHR$
      (13) AND Z$ <  >  CHR$ (27) THEN 2160
2170  IF Z$ =  CHR$ (13) THEN  NEXT I
2180  IF I > CASE THEN  RETURN
2190  VTAB 23: HTAB 1: CALL  - 958: INVERSE : PRINT "CHANGE DATA FOR VARIABLE # (1-";NVAR;")? ";: NORMAL : INPUT "";V
2200  IF V < 1 OR V > NVAR THEN 2190
2210  VTAB 23: HTAB 1: CALL  - 958: INVERSE : PRINT "PRESENT VALUE FOR CASE #";I;" ";NAME$(V);" = ";A(I,V): PRINT "CHANGE TO?
      ";: NORMAL : INPUT "";A(I,V)
2220 I = I - 1: NEXT I
2230  RETURN
2240  REM ******************
2250  REM * SELECTED CASES *
2260  REM ******************
2270  HOME : INVERSE : PRINT "CASE # TO REVIEW (OR 0 TO END)?";: NORMAL : INPUT " ";C1
2280  IF C1 = 0 THEN  RETURN
2290  IF C1 < 1 OR C1 > CASE THEN  PRINT : PRINT  CHR$ (7); CHR$ (7);"** CASE MUST BE BETWEEN 1 AND ";CASE;"***": FOR I = 1 TO
      1500: NEXT I: GOTO 2270
2300  HOME : INVERSE : PRINT " CASE # ";C1: NORMAL : PRINT : PRINT
2310  FOR J = 1 TO NVAR
2320  PRINT NAME$(J); TAB( 8)"= ";A(C1,J):
2330  IF NVAR >  = J + 10 THEN  PRINT  TAB( 21)NAME$(J + 10); TAB( 28)"= ";A(C1.J + 10): GOTO 2350
2340  PRINT
2350  IF J < 10 THEN  NEXT J
2360 J = NVAR
2370  VTAB 24: HTAB 1: CALL  - 958: INVERSE : PRINT "  <ESC> TO EDIT: <RETURN> TO CONTINUE  ";: GET Z$: NORMAL : IF Z$ <  >  CHR$
      (13) AND Z$ <  >  CHR$ (27) THEN 2370
2380  IF Z$ =  CHR$ (13) THEN 2270
2390  VTAB 23: HTAB 1: CALL  - 958: INVERSE : PRINT "CHANGE DATA FOR VARIABLE # (1-";NVAR;")? ";: NORMAL : INPUT "";V
2400  IF V < 1 OR V > NVAR THEN 2390
2410  VTAB 23: HTAB 1: CALL  - 958: INVERSE : PRINT "PRESENT VALUE FOR CASE #";I;" ";NAME$(V);" = ";A(C1.V): PRINT "CHANGE TO?
      ";: NORMAL : INPUT "";A(C1,V)
2420  GOTO 2300
2430  REM **********************
2440  REM * (1) DISK READ ERROR *
2450  REM **********************
2460 ERR =  PEEK (222)
2470  IF ERR = 5 OR ERR = 6 OR ERR = 11 THEN  PRINT D$;"DELEJE";F$: HOME : FLASH : PRINT "FILE DOES NOT EXIST. SELECT AGAIN...
      ...": NORMAL : PRINT D$;"CATALOG": FOR Z9 = 1 TO 10000: NEXT Z9: GOTO 1660
2480  IF ERR = 13 THEN  HOME : FLASH : PRINT "WRONG FILE TYPE.  SELECT AGAIN........": NORMAL : PRINT D$;"CATALOG": FOR Z9 =
      1 TO 10000: NEXT Z9: GOTO 1660
2490  GOTO 2660
2500  REM ***********************
2510  REM * (2) DISK WRITE ERROR *
2520  REM ***********************
2530 ERR =  PEEK (222): HOME
2540  IF ERR = 4 THEN  FLASH : PRINT "DISK IS WRITE-PROTECTED. INSERT ANOTHER": NORMAL : GOTO 1890
2550  IF ERR = 9 THEN  PRINT D$;"DELETE";F$: FLASH : PRINT "DISK FULL.  INSERT ANOTHER AND TRY AGAIN": NORMAL : GOTO 1900
2560  IF ERR = 10 THEN  FLASH : PRINT "FILE ALREADY EXISTS AND IS LOCKED.....": INPUT "WRITE OVER IT (Y/N)? ";A$: IF A$ <  >
      "Y" AND A$ <  > "N" THEN 2560
2570  IF ERR = 10 THEN 2600
2580  IF ERR = 11 THEN  FLASH : PRINT "ILLEGAL FILENAME.  SELECT AGAIN.......": NORMAL : PRINT D$;"CATALOG": INPUT "NEW FILE
      NAME: ";F$: HOME : GOTO 1900
2590  GOTO 2660
2600  IF A$ = "Y" THEN  PRINT D$;"UNLOCK";F$: GOTO 1900
2610  IF A$ = "N" THEN  INPUT "NEW FILE NAME (RETURN TO ABORT): ";F$: IF F$ =  CHR$ (13) THEN  HOME : END
2620  HOME : GOTO 1900
2630  REM ***********************
2640  REM * (3) ALL OTHER ERRORS *
2650  REM ***********************
2660  PR# 0:ERR =  PEEK (222)
2670  IF ERR = 254 THEN  PRINT : FLASH : PRINT "ILLEGAL INPUT...PLEASE TRY AGAIN.": NORMAL : RESUME
2680  HOME : PRINT "ERROR #";ERR;" AT LINE "; PEEK (218) +  PEEK (219) * 256;".": END
```

If the error code is 254 ("Bad Response to an Input Statement"), a short error message is flashed on the screen and you are given a chance to repeat the entry. This is intended to keep a bad data entry (for example, a number like 3L5 instead of 315) from abruptly ending the program.

If any other type of error is detected, the program prints the error code and the line at which it occurred. I'm averse to disabling the control-C. As long as it remains intact (as it does in Data Description), you can abort the program without turning off the machine or pressing reset.

If the initial data set is read from a disk file (routine beginning at line 1650), all errors are directed to the routine at 2460. Four main types of

ALL APPLES® ARE CREATED EQUAL. We Just Make Some Apples More Equal Than Others.

Make your Apple II equal to whatever task you might choose.

You've had your Apple for a while and you're beginning to cast covetous looks at those new, more expensive personal computers with access to more software packages. Or maybe you're getting ready to buy your first computer and you really like the Apple, but you also like some of the features and programs of those more expensive models.

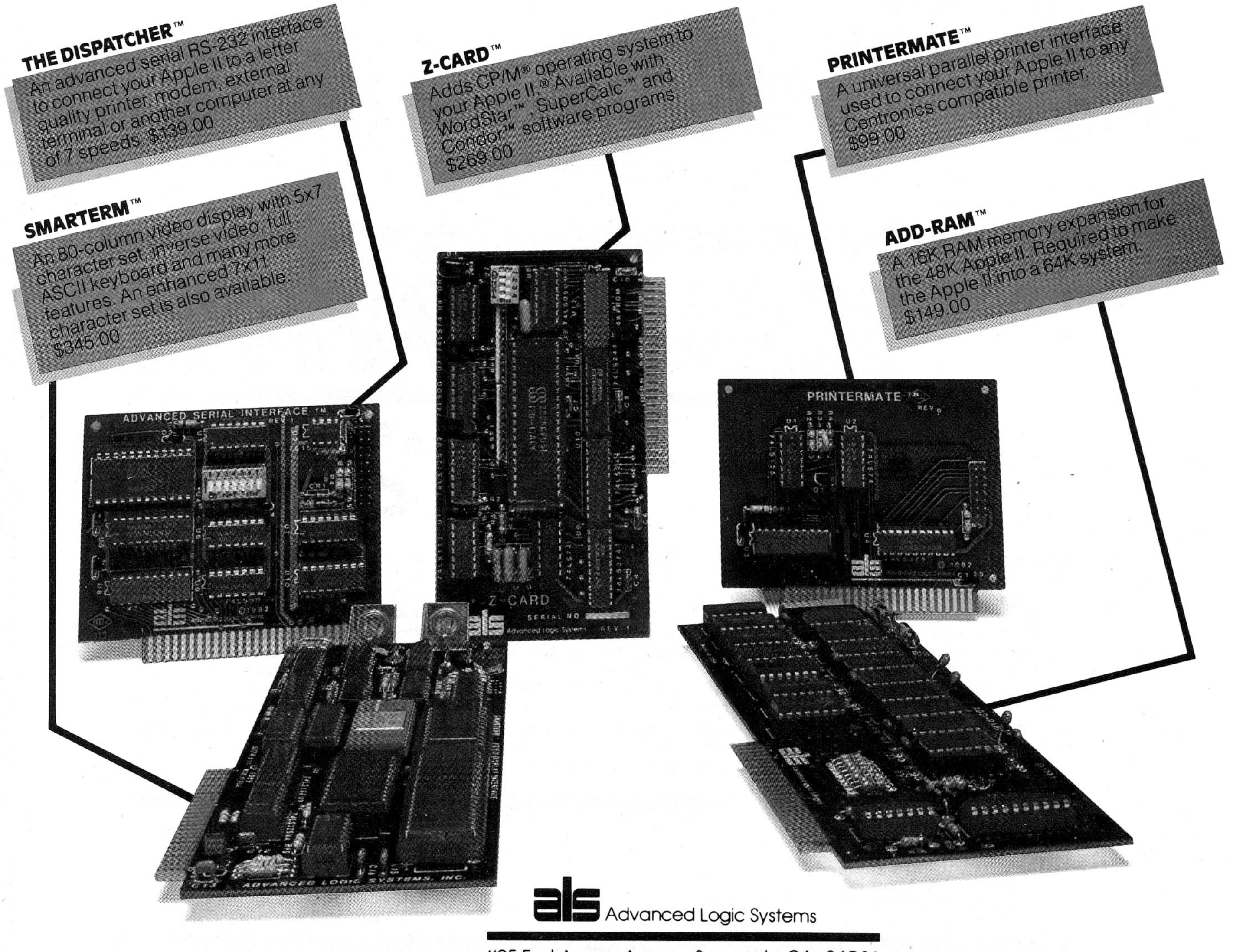
Good. Now's the time for you to know about Advanced Logic Systems. We're the people who put together all the plug-in function boards and popular software programs that make your Apple perform like the more expensive computers. At a fraction of the cost!

Features that exceed your expectations but not your budget.

All of the products from Advanced Logic have been designed with you in mind. We've made them easy to install—easy to operate. And we make the decision to purchase easy. Because we package the popular software programs, like WordStar, SuperCalc and Condor Jr. with the necessary hardware to run on your Apple II. These are the total-system packages like... The Words Set, The Numbers Set, The Files Set and the Synergizer Series. And, we stand behind all of our products with our 1 year limited warranty.

CALL TOLL FREE

Our customer service staff is waiting for you call—800-538-8177. We will be happy to send you our complete catalog of hardware and software packages.



als Advanced Logic Systems

1195 East Arques Avenue, Sunnyvale, CA 94086

Apple, Apple II are registered trademarks of Apple Computer, Inc. CP/M is a registered trademark of Digital Research, Inc. WordStar is a trademark of MicroPro International. SuperCalc is a trademark of Sorcim Corp. Condor is a trademark of Condor Computer Corp. Z-CARD, SMARTERM, ADD-RAM, THE DISPATCHER, PRINTERMATE and SYNERGIZER are all ours!

Circle 22 on Reader Service card.

read-related errors are screened: (1) END OF DATA—often the result of opening and trying to read a nonexistent file; (2) FILE NOT FOUND; (3) SYNTAX ERROR—illegal filename; and (4) FILE TYPE MISMATCH—trying to read data from any file type other than a text file.

Since most errors occur after the Apple has opened the file (often one that originally did not exist), the erroneous file is deleted and the user's disk catalog is displayed so the correct filename can be selected. The program then continues normally.

After the disk read has been successfully completed, the Onerr Goto is reset in line 1780 to the main error-trapping routine at 2660. This routine remains in effect until all calculations and printing have been completed, and the user is asked if he/she wishes to save the data set to disk (line 1520).

If so, the program moves to the disk save routine at 1870 and Onerr is set to 2530. Four types of write-related errors are screened: (1) WRITE PROTECTED—the write-protect tab is covered on the disk; (2) DISK FULL; (3) FILE LOCKED; and (4) SYNTAX ERROR—illegal filename. As with disk read errors, the display then tells you how to correct the error.

Printer Modifications

In general, you need change only a few lines to allow Data Description to run with an 80-column printer—some pokes, the length of the histogram strings and the Printer On command.

```
120  IF LEN(TITLE$)>65 THEN 110
1160 PR#1: PRINT CHR$(9) + "80N"
1330 DOT = CT(I): IF CT(I)>33 THEN DOT
=33
1490 PRINT: FOR I = 1 TO 80: PRINT "-";:
NEXT I: PRINT: PRINT
1580 ...POKE 36,79...
1590 ...POKE 36,74...
1600 ...POKE 36,71: PRINT DT$
```

To create a screen-only version of Data Description 2.0, you have two choices: eliminate the histogram routines, or modify the program so that the histogram prints on the screen separately from the statistics (as the second page for each variable).

For those of you with "math anxiety," perhaps access to a program like this will help ease the pain in your future dealings with numbers. An examination of the descriptive statistics for your data set is an essential first step in getting a feel for what it all means. Once you are comfortable with these simple statistics, you can begin to explore regression, analysis of variance, T-tests, factor analysis... ■

References

Nie, N., Hull, C., Jenkins, J., Steinbrenner, K. and Bent, D. *SPSS: Statistical Package for the Social Sciences*, 2nd Edition. New York: McGraw-Hill Book Company, 1975.

THE EASY TO USE, POWERFUL APPLE II WORD PROCESSOR!

Fastest by Far – *Compare speed to any other Apple II word processor . . .*

- Text editing speed rivals that of larger computers
- Extremely rapid text formatting
- Instant switching between editor and formatter with language card
- Rapid loading of standard text files

Easy to Use – ***designed with YOU in mind . . .***

- Easy to learn – includes interactive Tutorial
- Easy to remember, abbreviated commands
- Cut/Paste/Replicate with ease
- Adjustable automatic word wrap
- Editor remembers document name for easy disk updates

Powerful, Professional Features – ***won't soon be outgrown . . .***

- Auto-paragraph, headers, footers, test-page, user-definable page number format, centering, full margin control at all times, incremental spacing, left and/or right justify, tabs and tab stop setting, many more formatting features!
- Edit Macro – much more powerful than a mere "search and replace"!
- File chaining via unique "CALL" feature allows unattended printing of very large documents.
- Form Letter/Mail Merge – ideal for generating form letters, standard legal documents, bulk printing of letters or customized advertisements, begin/repeat blocks, and much more!
- Conditional Text with logical operators adds even more flexibility to Form Letter/Mail Merge feature.



WRITE AWAY

– AN ADVANCED WORD PROCESSOR –

by

Doug Stinson

Here's what some of our customers have to say about our word processor:

– *"This is my third (and by far the best) word processing system. This will replace over $300 in other software packages."* – ***J. B., Fenton, Missouri***

– *"The best word processor package for the Apple II that we have evaluated. . . Congratulations on a fine job of software development."* – ***A. C., Charlottesville, Virginia***

– *"Congratulations! I've thoroughly explored it and it's a very fine piece of work - extremely powerful and, despite my best efforts, virtually crashproof."* – ***J. B., St. Louis, Missouri***

– *"I love your word processor"* – ***H. S., Denver, Colorado***

Versatile – ***fits a wide variety of systems and applications . . .***

- Normal shift key operation for case shifting (optional).
- Can be used with or without lower case adapter.
- Supports optional Videx 80 column card (ideal for screen preview).
- Fast editor/formatter switching with RAM card installed.
- Supports Qume, NEC, Diablo, C. Itoh, Vista, Epson, many other printers.
- Uses standard text files – compatible with a wide range of other software, such as data base managers, user-written BASIC programs, and spelling dictionaries!

VISA | master charge

Demo Disk Available $15.00
Complete System Capability Demo to Give You the Ability to Use the System Before You Purchase. Credited On Full Purchase.

Complete System $175.00
Includes Full System Plus Extensive Users Manual.

Midwest Software Associates • **P. O. Box 301 • St. Ann, Missouri 63074**

Phone: *Toll Free* 1-800-362-2421/Ext. 467   MISSOURI: 1-800-835-2246/Ext. 467

Circle 50 on Reader Service card.

Trip Trapping

A handy, and readily accessible, error-trapping unit for your Pascal System.Library

by James R. Florini

The Pascal language design emphasizes carefully structured programming and rigidly specified variable typing. Unfortunately, this approach leaves all responsibility for entry error protection to the programmer. This "feature" (i.e., bug) of the language causes a lot of agony as Pascal programmers advance to the stage of writing interactive programs for others. It's very hard to anticipate all of the ways in which a naive user might misinterpret your carefully designed prompt statements. For example, if you've coded a program with the reasonable premise that *nobody* has a name longer than 40 characters, and thus declare NAME: STRING[40] as a variable, a longer name entry will be truncated to 40 characters with no warning to the user.

Another problem can result when the user makes an unexpected choice from a menu. Suppose you ask the user to reply with either a 1, 2, 3 or 4 to a question; you've carefully planned the actions the program will take following each of these, using that delightful feature of Pascal, the CASE statement. What happens if the user accidentally presses E instead of 3? In UCSD Pascal, the program simply skips the CASE statement entirely, and that is certainly *not* what you had in mind when writing the program.

Probably the most annoying "feature" of Pascal variable type restrictions is the fatal run-time error that occurs when a nonnumeric character is touched as the first "digit" when input of a real or integer is specified. ("Fatal" may sound extreme, but it really *feels* that way when the program stops—with loss of all entered data.) At first, it may seem that the best way to have a user enter an integer variable designated NUMBER is with a READLN (NUMBER). Nothing could be farther from the truth! If that user accidentally hits a nonnumeric key first, the whole program bombs. And if this is the 200th entry in a long series of data from a complicated experiment, he might say one or two uncomplimentary things about the programmer, Apple, UCSD and Blaise Pascal.

"What's needed is a system of error-trapping routines..."

What's needed is a system of error-trapping routines to prevent the user from making any entry not foreseen by the programmer—or at least keep the program from processing erroneous entries. Of course, omniscience is in relatively short supply, even among Pascal programmers, and the students who use my programs find some remarkably inventive interpretations for my prompts. Bitter experience indicates that the best solution is a series of entry routines that not only trap errors, but inform the user about them and suggest what the correct entries should be at that point in the program.

These routines are useful in virtually any program involving interactive entry of character, numerical or string data, so it is logical to include them in the System.Library where they can be used conveniently in any program. This Library is set up for convenient expansion to fit the user's needs. Like too much of the Apple documentation, the instructions in the manuals are rather imposing the first time they are encountered, and inexperienced programmers might be reluctant to make modifications to the Library. This article intends to show how a useful Unit can be installed in the Pascal System.Library. To make things com-

Address correspondence to Dr. James R. Florini, Biology Dept., Syracuse University, Syracuse, NY 13210.

fortable, I have presented a step-by-step description of exactly what must be done to include the Unit in the Library (see Listing 1). Finally, there is a short program (Listing 2) which illustrates how these utilities can be used in a program.

First, the Unit itself. The GET CHAR function is reasonably simple; it prints one or two prompt lines (specified when the function is called in the program), uses UNITCLEAR to make sure that the user doesn't respond until he sees the prompt(s), allows a chance to exit the program using the escape key, and converts any lowercase entries to uppercase (this minimizes the number of "acceptable" characters that must be specified in the program). If the character entered is not among those specified as the Function is called, the user is notified, shown a list of the acceptable characters that were specified at that point in the program, and prompted again. Only an acceptable response is returned to the main program. There's no way the user can make an unexpected response under these circumstances.

PROCEDURE GET ENTRY is a similar utility for the entry of strings. In this case, the length of the string is specified to avoid unexpected loss of characters from an overlong string. In keeping with my desire to let the user see just what's wrong when an entry error is detected, this procedure indicates precisely how much too long the string was. If you have the 1.1 version of Apple Pascal, you'll have to start program texts with the (*$V-*) compiler directive. This allows use of this part of the unit with strings of a variety of lengths. Although page 6 of the Addendum to the Language Reference Manual threatens dire consequences (i.e., "unpredictable results") if this string length checking is turned off, I've done this in many programs with no apparent ill effects.

The VALUE function does the same thing as the VAL function in Basic; it converts a string entry into a real (or, if ROUND or TRUNC is used, an integer) variable. This avoids all those infuriating program interruptions which result from nonnumeric entry errors, and also allows backspacing to correct errors the user notices before he presses the return key. The essential part of

Program listing 1. *Library modification functions and procedures.*

```
(*$S+*)
UNIT ENTRIES; INTRINSIC CODE 26;

INTERFACE
  TYPE RIGHT_ONES = SET OF CHAR;
  FUNCTION GET_CHAR (PROMPT1, PROMPT2:STRING; ACCEPTABLE: RIGHT_ONES): CHAR;
  PROCEDURE GET_ENTRY (LEN: INTEGER; VAR ENTRY: STRING);
  FUNCTION VALUE (ENTRY: STRING): REAL;

IMPLEMENTATION

FUNCTION GET_CHAR; (*Parameters and result type declared in INTERFACE*)
VAR CH, ENTRY: CHAR;
BEGIN
  REPEAT
    WRITE (PROMPT1);
    IF PROMPT2 <> '' THEN BEGIN WRITELN; WRITE (PROMPT2); END;
    UNITCLEAR (1);                        (*Clear type-ahead buffer*)
    READ (ENTRY); IF ENTRY = CHR(27) THEN EXIT (PROGRAM); (*If <ESC> entered*)
    IF ENTRY IN ['a'..'z'] THEN ENTRY := CHR (ORD (ENTRY) - 32);
    IF NOT (ENTRY IN ACCEPTABLE) THEN
    BEGIN
      WRITELN; WRITELN; WRITELN;
      WRITELN ('"',ENTRY,'" is not an acceptable entry.', CHR(7));WRITELN;
      FOR CH:=CHR(33) TO 'Z' DO IF CH IN ACCEPTABLE THEN WRITE ('"',CH,'", ');
      IF ' ' IN ACCEPTABLE THEN WRITE (' and <SPACE> ');
      WRITELN (' are appropriate entries at this point.'); WRITELN;
    END;
  UNTIL ENTRY IN ACCEPTABLE; WRITELN;
  GET_CHAR:=ENTRY;
END; (*get-char*)

PROCEDURE GET_ENTRY; (*See INTERFACE for parameter declarations*)
VAR I: INTEGER;
    SPACE: CHAR;
BEGIN
  REPEAT
    (* The next two lines draw a blank of length LEN between brackets, put the
     cursor at the begining of the blank, and read the entry.*)
    WRITELN; WRITE ('['); FOR I := 1 TO LEN DO WRITE ('_'); WRITELN (']');
    WRITE (' ',CHR(31)); READLN (ENTRY);  (*CHR(31) is "move cursor up"*)
    IF ENTRY<>'' THEN
    BEGIN
      IF (LENGTH (ENTRY)=1) AND (ENTRY[1]=CHR(27)) THEN EXIT (PROGRAM);
      IF LENGTH (ENTRY) > LEN THEN
      BEGIN
        WRITELN (CHR(7),ENTRY, ' IS ',LENGTH(ENTRY)-LEN,' TOO LONG!');
        WRITELN ('PRESS <SPACE> AND MAKE A SHORTER ENTRY.');
        READ (SPACE); WRITELN;
      END;
    END ELSE ENTRY:='';
  UNTIL LENGTH (ENTRY) <= LEN;
END;          (*get-entry*)
FUNCTION VALUE; (*See INTERFACE for parameter declaration, result type*)
VAR I, COMMA, NUMDIGITS, POINT, POWER, MAGNITUDE: INTEGER;
    STOREENTRY:STRING[10];
    DIGIT: STRING[1]; (*NOT the same as a CHAR, no matter what some books say*)
    NUMBER: PACKED ARRAY [1..10] OF INTEGER;
    LESSTHAN1, EXPONENT, NEGATIVE: BOOLEAN;
    DENOM, TEMP: REAL;

  PROCEDURE SHOW_ERROR;
  BEGIN
    WRITELN (CHR(7),STOREENTRY,' has a magnitude of ',MAGNITUDE,
    ', which is outside the acceptable range.');
    WRITELN ('Enter a correct number (with a magnitude between +37 AND -37)');
    GET_ENTRY(10,ENTRY); VALUE:=VALUE(ENTRY); EXIT(VALUE);
  END;

  PROCEDURE CHECK_LENGTH; (*Avoid value range error with NUMBER array*)
  VAR L: INTEGER;
  BEGIN
    REPEAT
      L:=LENGTH (ENTRY); IF L > 10 THEN
      BEGIN
        WRITELN (CHR(7),'The entry is ',L-10,' digits too long.');
        WRITELN ('Make a shorter entry.'); GET_ENTRY(10,ENTRY);
      END;
    UNTIL L<=10;
  END; (*check-length*)

  PROCEDURE CHECK_NUMERALS;  (*Check for non-numeric characters*)
  VAR I: INTEGER;
  BEGIN
    FOR I:=1 TO LENGTH (ENTRY) DO
    BEGIN
      IF NOT (ENTRY[I] IN ['0'..'9', '.', '-', '+', ',', 'E']) THEN
      BEGIN
```

Listing 1 continued.

Listing continued.

```
      WRITELN (ENTRY,' contains a non-numeric character, ',ENTRY[I],'.');
      WRITE (CHR(7), 'Please enter a corrected number. ');
      GET_ENTRY(10,ENTRY);  (*Length of ENTRY must be <= NUMBER array*)
      CHECK_NUMERALS; EXIT (CHECK_NUMERALS);
    END;
  END;
END;

PROCEDURE GET_EXPONENT;
VAR E, L: INTEGER;
    NUM: STRING[5];
BEGIN
  EXPONENT:=TRUE;E:=POS('E',ENTRY);
  LESSTHAN1:=COPY (ENTRY,E+1,1)='-'; (*Determine if negative exponent*)
  L:=LENGTH(ENTRY); NUM:=COPY(ENTRY,E+1,L-E);
  POWER:=ROUND(ABS(VALUE(NUM))); DELETE (ENTRY,E,L-E+1);
  IF POINT<>0 THEN MAGNITUDE:=POINT-1
     ELSE MAGNITUDE:=LENGTH(ENTRY)-1;
  IF LESSTHAN1 THEN MAGNITUDE:=MAGNITUDE-POWER
       ELSE MAGNITUDE:=MAGNITUDE+POWER;
    IF (MAGNITUDE > 37) OR (MAGNITUDE < -37) THEN SHOW_ERROR;;
  END;

BEGIN   (*main value function*)
  IF ENTRY = '' THEN
  BEGIN
    WRITELN (CHR(7),'NO ENTRY WAS MADE!'); VALUE:=0; EXIT (VALUE);
  END;
  CHECK_LENGTH; CHECK_NUMERALS; STOREENTRY:=ENTRY; NEGATIVE:=ENTRY[1]='-';
  POINT := POS ('.', ENTRY);
  IF POINT <> 0 THEN DELETE (ENTRY, POINT, 1); (*remove decimal point if there*)
  IF POS('E',ENTRY)<>0 THEN GET_EXPONENT ELSE EXPONENT:=FALSE;
  REPEAT
    COMMA := POS (',', ENTRY);
    IF COMMA <> 0 THEN DELETE (ENTRY, COMMA, 1);
  UNTIL COMMA = 0;
  NUMDIGITS := LENGTH (ENTRY);
  (*>>>>>> The following 4 lines are the central part of VALUE <<<<<*)
  FOR I := 1 TO NUMDIGITS DO
  BEGIN
    DIGIT := COPY (ENTRY, I, 1); NUMBER[I] := POS (DIGIT, '123456789');
  END;
  TEMP := 0; FOR I := 1 TO NUMDIGITS DO
    TEMP := TEMP + NUMBER[I] * PWROFTEN (NUMDIGITS - I);
  IF POINT <> 0 THEN DENOM := (PWROFTEN (NUMDIGITS - POINT + 1))
     ELSE DENOM := 1;
  TEMP := TEMP /DENOM; IF NEGATIVE THEN TEMP := -1 * TEMP;
  IF EXPONENT THEN
  BEGIN
     IF LESSTHAN1 THEN
     BEGIN
       WHILE POWER > 38 DO     (*Avoid out of range error*)
       BEGIN
         POWER:=POWER-1; TEMP:=TEMP/10;
       END;
       TEMP:=TEMP/PWROFTEN(POWER) (*Negative exponent*)
    END ELSE TEMP:=TEMP*PWROFTEN(POWER);    (*Positive exponent*)
  END;
  VALUE := TEMP;
END;  (*value*)

END.  (*This indicates the end of the Unit; no BEGIN is necessary.*)
```

the Function is the analysis of the string of numerals, digit by digit; an integer value is assigned to each digit by reading its position in the string 123456789. The rest of the Function takes care of negative and exponential entries, checks that the string isn't too long, that all entries are numerical, goes to some trouble to prevent out-of-range errors, and even removes commas in case the user isn't aware that 12,345 is not compatible with Apple's handling of numerical variables. To take advantage of feedback to the user on out-of-range or nonnumeric errors, it's a good idea to use the VALUE function immediately after a string is entered, not after a whole series of entries. Besides, real or integer variables use much less memory than do the corresponding strings.

Note that recursion is used in three of the procedures in the VALUE function. CHECK_NUMERALS keeps calling itself until all the entries are in the specified set, and GET_EXPONENT and SHOW_ERROR both call the entire VALUE function (of which they are parts) to get the value of the exponential portion of the Entry string. Also, note that one part of the Unit (VALUE) can call another part (GET_ENTRY) of the same Unit.

Cookbook Instructions

Now, how do we get all these nice things included in the System.Library?

1. Enter and compile UNIT UENTRIES just as printed here. Save it as UENTRIES on your boot disk, and on a backup disk just to be safe.

2. Save the current System.Library onto a backup disk as Old.Library. These instructions are quite foolproof, but...

3. Insert the Apple 3: disk in drive #2. From the Command: level, X)ecute APPLE3:LIBRARY.

4. The first thing that appears is the prompt "Output code file ->", which means "What are you going to call the new Library?" Enter SYSTEM.LIBRARY, or * for short. Then press return.

5. The next prompt, "Link code file ->", means "What library is the lucky recipient of the new unit?" Again, enter either SYSTEM.LIBRARY or *.

6. Now the screen fills with columns of numbers, unit names, etc., and the top line shows a remarkably incomprehensible prompt. If you have a 40-character screen, you can use control-A to see the other half of the screen, but there's nothing of much interest there. Enter = because you want to keep everything that's in the current Library; no return is necessary.

7. There is now much whirring, lots of flashing, and the names of all the Units currently in the System.Library are transferred to the list of the new library on the lower half of the screen. It's rather exciting the first time you see it. For the Pascal 1.0 system, five slots are filled. For 1.1, the first six slots are occupied (Chainstuff was added).

8. When the action stops, enter N to show that you want to enter a new Unit. The prompt "Link code file ->" appears near the top of the screen.



CHRISTMAS SEASON SPECIALS!

Let ARK COMPUTING Make This Your Best Christmas Ever!

Super Fan II by R.H. Electronics **59.95**/79.95

Applicard, a high performance Z-80 card with 64K Ram, complete with CP/M
4 mhz **324.95**/445.00
6 mhz **395.00**/595.00

Microsoft Z-80 card with CP/M and Microsoft Basic
2 mhz **269.95**/395.00

Microtek Parallel Printer Interface complete with centronic compatible connector
64.95/79.95

Lazer Lower Case +Plus with Character Set +Plus **49.95**/84.90
Lower Case +Plus alone **39.95**/59.95

Lazer Graphics +Plus **99.95**/159.95
Graphics +Plus and Lower Case +Plus **134.95**/219.90

Computer Stop 16K Ram Board **69.95**/149.95

Computer Stop Omnivision 80 Column board
129.95/295.00

Videx Video-term with softswitch, inverse character set and 80 column Visicalc preboot
295.00/450.00

Wizard BPO 16K buffered printer interface (expandable to 32K) **134.95**/179.95

Wizard 80, 80 column board **195.00**/295.00

Lazer Pascal **29.95**/39.95

Anix 1.0 **34.95**/49.95

Lazer Forth **44.95**/59.95

D Tack 68000 board for the Apple II with 4K Ram **895.00**

Lazer Model/32 (16032 board for the Apple II)
CALL!

Lisa	**59.95**/79.95
Lisa Educational Pak	**79.95**/119.95
Alien Ambush	**19.95**/29.95
Bandits	**19.95**/29.95
Cannonball Blitz	**24.95**/34.95
County Fair	**19.95**/29.95
Cranston Manor	**24.95**/34.95
Cyclod	**19.95**/29.95
David's Midnight Magic	**24.95**/34.95
Dosource 3.3	**24.95**/39.95
Dueling Digits	**19.95**/29.95
Falcons	**21.95**/29.95
Firebird	**21.95**/29.95
Foosball	**19.95**/29.95
Horizon V	**25.95**/34.95
Genetic Drift	**19.95**/29.95
Kabul Spy	**24.95**/34.95
Jelly Fish	**19.95**/29.95
Lemmings	**19.95**/29.95
Labyrinth	**19.95**/29.95
Mouskattack	**24.95**/34.95
Outpost	**19.95**/29.95
Red Alert	**19.95**/29.95
Pig Pen	**24.95**/34.95
Russki Duck	**25.95**/34.95
Minator	**24.95**/34.95
Track Attack	**19.95**/29.95
Thief	**17.95**/29.95
Space Quarks	**19.95**/29.95
Snack Attack	**19.95**/29.95
Swash Buckler	**24.95**/34.95
Gin Rummy	**24.95**/34.95
The Dictionary	**69.95**/99.95
General Manager	**99.95**/149.95
4 Ft. Disk Cable	**19.95**/29.95
Visicalc	**179.95**/250.00
Using 6502 Assembly Language Book	**14.95**/19.95
Kids and The Apple Computer Book	**15.95**/19.95
Apple Panic	**19.95**/29.95
Kraft Joystick	**49.95**/69.95

ARK COMPUTING INC.

Your Salvation In The Sea Of Inflation.

714 735-2250
P.O. Box 2025
Corona, CA 91720



Circle 29 on Reader Service card.

Program listing 2.
Program test.

```
PROGRAM TEST_UNIT;
USES ENTRIES;
VAR ENTRY: STRING;
    NUMBER:REAL;
    CHOICE:CHAR;

BEGIN {MAIN TEST_UNIT}
  REPEAT
    WRITELN ('Enter a string of numbers to be evaluated:');
    GET_ENTRY(10, ENTRY); NUMBER:=VALUE(ENTRY);
      WRITELN ('THE NUMBER IS ',NUMBER,' OR ',NUMBER:8:6);
      WRITELN ('   WHICH IS TWO TIMES ',NUMBER/2:8:6);
      WRITELN ('   AND ROUNDS TO ',ROUND (VALUE (ENTRY));
      WRITELN; WRITELN;
    CHOICE:=GET_CHAR ('ENTER ANOTHER NUMBER? ',' ',['Y','N']);
  UNTIL CHOICE='N';
END.
```

9. Assuming you carefully followed instruction 1, now enter UENTRIES. After more whirring and flashing, the upper half of the listing will disappear and you'll see

1-(26) ENTRIES 2038

in the upper part. If this doesn't happen, check again to be certain that you have UENTRIES.CODE on the boot disk, and that you haven't changed the Prefix disk designation.

10. Now type 1 (the number of the slot where Entries is listed) and press the space key (not return) followed by the number of the first empty slot listed at the bottom of the screen. (For Pascal version 1.0, this should be 6, and for 1.1 it should be 7.)

11. More whirring and flashing. Your new unit advances to its proper station among the blessed, and the same old first-line prompt returns. Now type Q (absolutely *not* A! Why did they pick two adjacent keys?) to end things.

12. The Apple will beep as if you made a mistake, but you didn't. A "Notice?" prompt appears at the bottom of the screen. Now you can enter anything you want—a copyright notice, something to remind you about what you did here, or even a string of vulgar words. (The only way this message can be seen is by using the LIBMAP program, so you aren't likely to offend anyone.) End it by pressing return.

13. After a bit more whirring, the Command: prompt line reappears, and you now have a new System.Library which is five blocks longer than the original version and has a new date associated with it. Check by going to the F)iler and getting an E)xtended directory of the boot disk.

14. Enter and run Program Testunit to see if everything came out properly.

Now you can use these utilities in any program by just including USES ENTRIES; after the program name. The Testunit program illustrates their use and the way the parameters are specified. ■

Big Business or Bust

As part of inCider's policy to expose all areas of Apple computing to public scrutiny, here are personal reflections on the pros and cons of the computer mail-order business from a school principal who turned to computing to avoid headaches. . . and instead found migraines.

by Fred Huntington

Please pay attention! I'm about to tell you how to make a million in the computer mail-order business. If you follow my advice, you, too, can make a million. . . headaches, that is.

These are not the writings of someone with only a fleeting understanding of the subject. First-hand knowledge inspired this treatise. But why on earth would anyone get involved with something so crazy as mail-order?

Three years ago, I sold my home in San Diego and moved into the middle of a cotton field. Corcoran is the name of the town, halfway between Bakersfield and Fresno. En route, I bought my first Apple II in Los Angeles. I'd never *touched* a computer before, so I had no idea what "loading programs," "I/O error" or "booting" meant when the salesman discussed the sale with me.

I eagerly loaded my 32K byte Apple (with Integer Basic) and a cassette recorder into my car, and left the store. For six months, computer games and I became fast friends. I can still remember the excitement of Space Invaders—what a game! During this time I realized my Apple could help me in my role as a school principal. I programmed a simple but effective yard-duty schedule, which actually functioned properly. I was ecstatic. My success encouraged more programs—if I could write one, I could write more.

My second quest was the conversion of a test I had written five years earlier. I had great fun with that one, but the coding ultimately came out perfectly. My confidence was building.

My next project, if you can believe this, was a test that judged a user's pickiness. The inspiration for this wildly successful program was my wife's failure to completely butter the toast all the way to its edges one morning. I was becoming invincible (and insufferable). I could program anything at all!

I wrote several additional programs in the belief that, while the major publishers might be blind to my efforts, normal people would eagerly buy my creations. I placed an ad in my favorite magazine—one-sixth of a page, on a contract ad, was $192. For this price I had to provide the camera-ready layout. Believe me, $192 was quite an investment at that time.

In the meantime, I discovered that the nearest computer store that sold software was 200 miles from my home. I decided that, as long as I was selling my own programs, I might as well offer other publishers' programs too.

After 20 or 30 sales, I figured, I'd be in the big bucks. However, much to my shock, I found that the market for my Acanthopterygian Fortune-Telling and Chauvinism programs was less than clamorous. At least rock-bottom sales don't fall much lower than zero. I decided to focus on a single program, hoping that would spark a recovery. My second ad touted the Fortune-Telling program, since I *knew* this was the one item everyone wanted. The advertisement appeared, and I awaited the constant ringing of the phone. Not one copy sold!

Of such travails are the mail-order business made. Here are some of the lessons I've learned. First, allow yourself plenty of time for business planning. Also, commit to your understanding that there will be many, many late nights of hard work, filling orders and planning additional catalogs.

For the first year-and-a-half, my wife Barb and I worked out of our home. Barb was still teaching, and I had my outside job. This caused a definite problem. UPS couldn't deliver packages to our home during the day because no one was there. Without UPS, the business was dead.

I decided to have UPS deliver the materials to me at the school where I

Address correspondence to Dr. Fred Huntington, 1945 South Dairy Ave., Corcoran, CA 93212.

worked. Unfortunately, this caused some difficulties with a large software firm. I placed my first order with this company, but they wouldn't ship it to me—they thought the buyer was a school trying to purchase the software at wholesale prices.

Eventually this was cleared up. The packages were delivered to the school, and I shipped five to ten orders per day from the same location. This was fine until I began to receive more and more orders, and UPS frequented the school more often than the students. The teachers...the community...everyone wondered what I was up to. What was in those packages leaving the school? Even worse, what was in the packages entering the school? Did the FBI know about my activities?

At about this time we decided someone should work full-time at the house. We were taking a risk, but success doesn't come without gamble. Besides, I certainly looked forward to a shorter work day—less work, more enjoyment.

Hah! After hiring our first full-time employee to answer the phones, a strange thing occurred. We got *more* orders, and were falling further behind and staying up even later than before. The firm needed another full-time employee.

Invoicing and inventory ended up computerized at this time. Armed with a Paper Tiger 460 and High Tech's Store Manager, our invoices began to look professional. Unfortunately, we didn't plan our data entry correctly, and had to spend more time entering special inventory items. We could have managed the entire inventory at once had we input our total inventory supply from the start.

As we became better and better at filling the flood of orders, we hired six more employees. These wonderful folks worked in our house when Barb and I left to go to our jobs. You've all seen wall-to-wall carpet—in our house it was wall-to-wall software. We had lost our privacy, so the next step was inevitable. We went to look for a place to rent for our business.

Old court houses, movie theaters, meat warehouses, auto shops...none proved suitable. Then we found an abandoned supermarket/ladies gym/barber shop that seemed appropriate, and there was room to expand. We began by setting up the business in a small corner of the supermarket. The total rent was $150. Should you find a rental at this price today, you must also be in Corcoran.

This little corner operation soon expanded into a big corner—then half the supermarket—and then all of the supermarket. The barber shop and ladies gym were eaten up by our expansion as well. Now, we're bursting at the seams. We have architectural renderings for a beautiful 13,000

"The advertisement appeared, and I awaited the constant ringing of the phone. Not one copy sold!"

square foot facility, complete with railroad spur, day care center, gazebo, health food store, and a myriad of other nifty additions, for some day in the future.

I was never accused of being too sane. And I thoroughly enjoy computers. These are the necessary and basic requirements for a successful mail-order business. Neither of us had a solid business or computing background, so we tried things that business folk insisted we shouldn't try—and found them successful. The financial side hurt us, however. Don't ever forget this following bit of advice: Hire a good bookkeeper, and keep all of your books in smart order.

As far as not knowing much about computers, this put Barb and me in the same boat as most of the people buying our software. Our enthusiasm for new products was genuine, and our sympathy for consumers sincere.

Problems that seem insurmountable at the time can usually be worked out. I remember one customer—a best-selling software author, no less—who walked into our store and purchased a VersaWriter, which he took with him when he left. Wouldn't you just know it? One of the employees thought his purchase was for a mail order, and sent him a second VersaWriter.

The author very nicely returned the item to us at his own expense, with only a few terse suggestions about the art of staying in business, one of the conditions being an end to free VersaWriters. When he received a third VersaWriter, I wrote him a letter explaining that the business was new, and that my embarrassment was genuine. It was the fourth shipment that permanently ended our relationship. The situation was so bad that I've blotted the details from my memory to preserve what sanity remains to me.

Another problem you'll face is the customer who wants a special item drop-shipped—shipped directly from the manufacturer or distributor.

For example, a gentleman from Colorado wanted a SilentType printer "blue-labeled" (a UPS two-day shipping method). This hardware was to be drop-shipped from our distributor, who agreed to the drop-shipment and stated the printer would go out UPS that day. I immediately notified the eager customer and charged his credit card with the purchase. Everything was just great—until I found out that the order never left the distributor. The customer was livid. We lost a valued customer who had originally wanted to buy from us because "we were small, and guaranteed everything."

"...you are never without something to play with on your computer."

The bane of the mail-order business is the price increase. With a three-month time lag from when we write the ads to when they're published, we're vulnerable to manufacturers who raise their prices with little or no warning. Sometimes you don't find out about the higher prices until you pay an invoice. So what do you do for the customer? Send the software at its higher price (which is certainly not the price the customer ordered the item at), or cancel the shipment and enrage the customer? Sometimes, it just doesn't pay to get out of bed.

One of the benefits of the mail-order business is that you are never without something to play with on your computer. Problem is, you have so much you never have time to play with any of your stock. You also meet some of the nicest people. Computer folk seem to be a special breed—friendly, warm and fun to be with. Of course, there are some jerks as well. I'd tell you about one of them, except I'm certain I'd end up being sued again.

If you have family, involve them from the start. My new son, Dale, is learning the computer business from the ground up. My daughter, Melody, is a computer freak just like her Mom and Dad. She loves Hodge Podge, but has shown some interest lately in Frogger.

I hope I've pointed out some of the pitfalls, and some of the joys, in the computer mail-order business. No matter what anyone says, "mail-order" doesn't have to be a dirty word. ■

A Galaxy of Great Games

the PROGRAM STORE

LOST COLONY



By David Feitelberg from Acorn

You are the Economic Manager of the world's first space colony. The next support ship from Earth isn't due for another **15 years**, and you have instructions to make things go better or get out of office in shame. You must allocate labor, explore new territories, decide on production quotas, determine pay scales and taxes for the most productivity. You're armed with maps and charts. 10 levels of difficulty; "save the game" feature on disk.

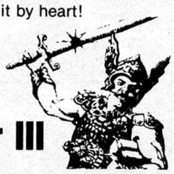
Atari: 40K Disk/ **IBM:** 48K Disk/ **TRS-80:** 16K Tape or 32K Disk, **$29.95**

MASTERTYPE

From Lightning

Imaginative and fun, this "space typing" game will teach you the keyboard in SELF DEFENSE! The only way you can defend yourself against the letters coming at you from all corners of the screen is to press each matching letter quickly. As the pace picks up, they're flying so fast, ***there's no time*** to look at the keyboard—you'll ***have to*** learn it by heart!

Apple: 48K Disk,
Atari: 32K Disk, **$39.95**
IBM: 64K Disk



ZORK I, II, or III

From Infocom

You can communicate ***in complete English sentences*** in these interactive adventures, with a vocabulary of over 600 words! Each of these literate games will keep you entertained for 50 hours or more. **Zork I,** The Great Underground Empire: Discover 20 treasures, and fight for your life! **ZORK II,** The Wizard of Frobozz: The Wizard will attempt to confound your quest with his capricious powers. **Zork III:** Brand New Adventure!

Apple: 32K Disk/**Atari:** 32K Disk/ **IBM:** 64K Disk, **$39.95 Each**

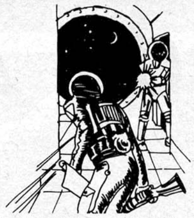


CRUSH, CRUMBLE AND CHOMP!

From Epyx

It's a monster movie, and ***you are the monster!*** You can be The Glob, Kraken, Mantra, Mechismo, Arachnis or Goshilla—or even design your own "custom" monster (disk version only). Hilarious action game, loaded with graphics and sound, as you practice your villany and avoid science, the police and the armed forces.

Apple: 48K Disk/**Atari:** 32K Tape or Disk/**TRS-80:** 16K Tape or 32K Disk, **$29.95**



VOYAGER I

From Avalon Hill

You're on board a spaceship infested with killer robots in this graphic science fiction game. You must clear the 4-level 144-location ship of robots and arm it to self-destruct. Can you do it and escape before you, too, are blown up? High-speed graphics are represented in 3-D perspective representing your eye's view; with instant switching to floor plan maps.

Apple: 48K Disk, **$24.95**
Atari: 32K Tape, **$19.95;** 32K Disk, **$24.95**
Color/TRS-80: 16K Tape, **$19.95**



CHOPLIFTER

From Broderbund
#1 Best Seller!

Can you rescue the hostages captured by the Bungelings? First you'll have to make careful plans for your helicopter rescue mission. But hurry—every minute counts! Who knows ***what*** the Bungelings have in mind? Requires joystick.

Apple: 48K Disk/**Atari:** 40K Disk, **$34.95**

SHOOTOUT AT THE OK GALAXY

From Avalon Hill

This exciting new game requires fast arcade response and well-thought-out strategy. ***Thirty alien warships*** have entered your Patrol Zone—can you handle your defense? Are your shields up? Have you checked your energy level? Is your azimuth set? OK then . . . ***Good Luck!***

Apple: 48K Disk, **$24.95**
Atari/Color/ TRS-80: 16K Tape, **$19.95**



CROSSFIRE

From On-Line

The most original, exciting arcade game—designed exclusively for the home computer—that we've seen! The aliens have taken over your city, and everyone left peaceably—***except you.*** You wander the streets, shooting at aliens firing at you ***from all 4 directions.*** You must move quickly to avoid their CROSSFIRE, gather bonus tokens, and recharge your laser. It's one of our favorites! Requires Joystick.

Apple: 48K Disk, **$24.95**
Atari: 16K Tape or 32K Disk, **$29.95;** ROM Cartridge (16K), **$44.95**
IBM: 64K Disk, **$29.95**



FROGGER

From On-Line; Cornsoft

One of the most popular arcade games ever . . . now available for your home computer! Just as in the arcade game, you must dodge heavy traffic and jump the moving logs in the stream to get home. Fantastic graphics! Requires joystick.

Apple: 48K Disk, **$34.95**
Atari: 16K Tape or 32K Disk, **$34.95**
TRS-80: 16K Tape, **$19.95;** 32K Disk, **$22.95**



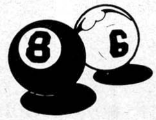
THRESHOLD

By Warren Schwader & Ken Williams

You, alone, can save the Earth Federation from the lawless aliens who roam our galaxy. Not only have these invaders been attacking our home planet—they're also raiding our space colonies! Space travel has become increasingly dangerous, as has friendly trade. But supplies must go through. As the pilot of the Federation fighter THRESHOLD, it's all up to you! High resolution graphics and sound plus "save the game" feature. Requires joystick.

Apple: 48K Disk/**Atari:** 40K Disk, **$39.95**
VCS: $22.95

POOL 400/ POOL 1.5



From IDSI

A remarkable action-simulation . . . it looks and plays ***just like the real thing!*** You can play several different types of pool at your choice of table speed with "instant replay" of any shot. Hi-res color graphics; requires joystick.

Apple: 48K Disk, **$34.95**
Atari: ROM Cartridge (16K), **$39.95;** 48K Disk, **$34.95**



WIZARD OF WOR

From Roklan

Can you defeat the WIZARD OF WOR? First you must descend into the ever-changing maze of Dungeons with your Worriors, and do battle with the monsters you encounter, like the Burwors, Garwors, Worluk and enemy Worriors. Only then can you turn your attention to the Wizard, who can teleport magically around the screen, hurtling lightning bolts as he moves. Simultaneous 1 or 2 player action.

Atari: 32K Disk, **$39.95;** ROM Cartridge (16K), **$44.95**
VCS

VISA **For Information Call 202-363-9797** **To Order Call Toll-Free 800-424-2738** **MasterCard**

the PROGRAM STORE **4200 Wisconsin Avenue, NW, Dept. 28-01-3 Box 9582 Washington, D.C. 20016**

Visit our other stores:
Seven Corners Center, Falls Church, VA • W. Bell Plaza, 6600 Security Blvd., Baltimore, MD • Olentangy Plaza, 829 Bethel Rd., Columbus, OH

THE PROGRAM STORE • Dept. 28-01-3 • Box 9582 • 4200 Wisconsin Avenue, NW • Washington, D.C. 20016

Item ______ Tape/Disk ______ Price ______ Postage $2.00
Total ______
☐ CHECK ☐ VISA
☐ MASTERCARD
Computer ______

Name ______
Address ______
City ______ State ______ Zip ______
Card # ______ Exp ______

rcle 38 on Reader Service card.

Speedy Alchemy

Build this simple pH meter interface for under $20—and give your lab students a little (comic) relief.

by Rolf A. Deininger and Carl F. Berger, Jr.

Data acquisition and reduction remains a largely unexplored microcomputer utility. Microcomputers are ideally suited to collect data from instruments, analyze it, and have it available in machine-readable form for transfer to other computers or for archival purposes.

A pH meter is a typical laboratory instrument. It measures the degree of acidity or basicity of a solution, and expresses the result in terms of pH, the negative logarithm of the molar hydrogen ion concentration. The pH scale ranges from 0 (very acidic) to 7 (neutral) to 14 (very basic). A pH meter consists of a probe that contains a hydrogen-ion-sensitive glass bulb and a reference electrode. A voltage differential directly proportional to the hydrogen-ion concentration develops at the probe; this differential is measurable and, after suitable amplification, is displayed on the voltmeter of the instrument panel. The voltmeter on the (somewhat antique) pH meter shown in the photo responds to a voltage between −25 and +25 mV (millivolts).

Interface

To interface the pH meter directly to an Apple II microcomputer without an A/D (analog-to-digital) converter (there are several on the market but they cost from $200–$500), you must convert the −25 to +25 mV signal into a resistance between about 100 and 150k ohms. This resistance can then be measured by one of the four simple circuits built into the Apple II that normally measure the resistance of the potentiometers in the paddles.

The Circuit

Figure 1 shows a diagram of a circuit that amplifies and shifts the voltage signal of the pH meter to a positive voltage and then converts the voltage into a resistance through the 2N2222 transistor. The circuit needs to be tuned to give a good response in the range between a pH of 4 and 10. The actual amplification circuit can be housed in a small box as shown in the photo.

Calibration

The circuit was calibrated using standard pH reference solutions of 4, 7 and 12. Table 1 shows the corresponding paddle readings. Although a plot of the data shows that the circuit is not quite linear, for purposes

Program listing. Apple Integer Basic program for displaying pH readings.

```
LIST 100,1440
 100 REM  INEXPENSIVE PH-METER TO APPLE INTERFACE AND DISPLAY
 110 REM  BY R.A.DEININGER AND C.F.BERGER AUGUST 1981
 120 GOTO 3000: REM PRINT TEXT PH AND START
 130 GR : COLOR=15: GOSUB 330
 140 REM DRAW DECIMAL POINT AND
 150 REM TAKE 100 SAMPLES AND AVERAGE THEM
 160 SPDL=0: VLIN 37,39 AT 20: VLIN 37,39 AT 21
 170 FOR I=1 TO 100:SPDL= PDL (0)+SPDL: NEXT I
 180 PD=SPDL/100: IF PD>210 OR PD<50 THEN GOTO 300
 190 REM  CONVERT PADDLE READING TO PH
 200 IF PD<150 THEN PH=(11500-30*PD)/100
 210 IF PD>=150 THEN PH=(14500-50*PD)/100
 220 REM  DRAW LEFT DIGIT
```

Listing continued.

Address correspondence to Rolf Deininger and Carl Berger, Jr., The University of Michigan, School of Public Health, Ann Arbor, MI 48109.

Circle 21 on Reader Service card.

MEGABYTERS
Make Apples Grow



SVA developed the First 8″ floppy controller in 1979, the First 256k RAM Card in 1981, and now introduces PRO-GUARD, the First 8″ floppy controller for Apple *III*.

We offer the most complete line of Memory Expansion Systems available and have a worldwide reputation for Reliability, Compatibility and Support. Every product undergoes a seven day burn in period and is retested prior to shipment.

Our standard Apple DOS*, CP/M* and Pascal* distribution software provide maximum application flexibility, and our full-time Customer Relations Department is ready to assist you with start-up and maintenance requirements. SVA means business. For more information call or write today.

SVA

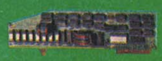
*Trademarks: CP/M-Digital Research, Apple Dos-Apple Computer, VisiCalc-VisiCorp.

SORRENTO VALLEY ASSOCIATES, 11722 SORRENTO VALLEY ROAD, SAN DIEGO, CA 92121-1084 TELEPHONE (714) 452-0101 TWX 910-335-2047



ZVX4

- 8″ Floppy Controller
- Single/Double Density Media
- Single/Double Sided Drives
- Automatic Density and Sidedness Detection
- IBM Standard Formats
- Corvus, Amlyn Compatible
- Autoboot DOS, Pascal, CP/M



APP-L-CACHE

- 64k, 128k, 192k, 256k models
- Expands VisiCalc* worksheet to 252k
- Built-in Language Card
- Needs only 1 slot
- Disk Emulator Mode
- Speed of Hard Disk
- Eliminates need for second mini-drive
- Low Power Consumption



PRO-GUARD

- Apple III 8″ Controller
- Increases online Storage up to 2.2 megabytes
- Backup Apple Profile
- DOS, SOS, Compatible
- IBM Standard Formats
- Under SOS, data transfer speed approaches hard disk



AMS 5000

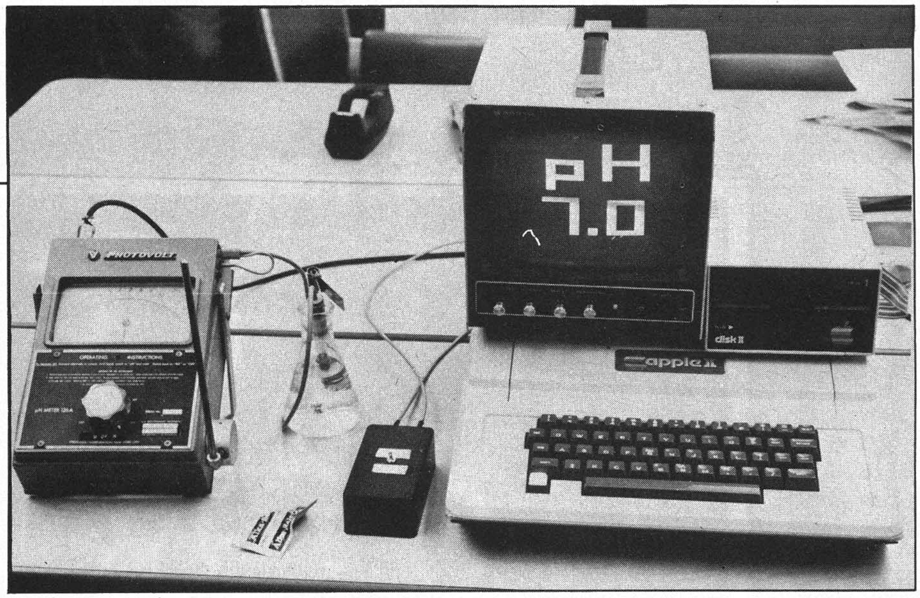
- 5¼″ Winchester Turnkey System
- 5, 10, 20 megabyte models
- Compatible with ZVX4 8″ Controller
- Network up to 127 Apples
- User selectable storage allocation for DOS, CP/M, Pascal
- Apple III compatible



AMS 8000

- Twin Drive Turnkey System includes ZVX4 Controller
- Interchange Data with other Computers
- Increase online storage up to 4.4 megabytes
- Use for Hard Disk Backup
- Includes DOS, CP/M, Pascal Distribution Software

Our pH meter and Apple II. From left to right: pH meter, beaker with probe, interface circuit in small box and Apple II.



of demonstration the pH value is calculated based on two straight-line approximations through the three points. The approximate formulas for the pH values are:

for pH < 7: 11.5 – 0.03 * PDL(0)
for pH > 7: 14.5 – 0.05 * PDL(0)

The Program

The program listing shows a small program in Integer Basic that displays the pH value in large letters on the screen for classroom viewing. Enough remark statements have been included to make it self-documenting. Most of

pH	Paddle (0)
4	210
7	150
10	50

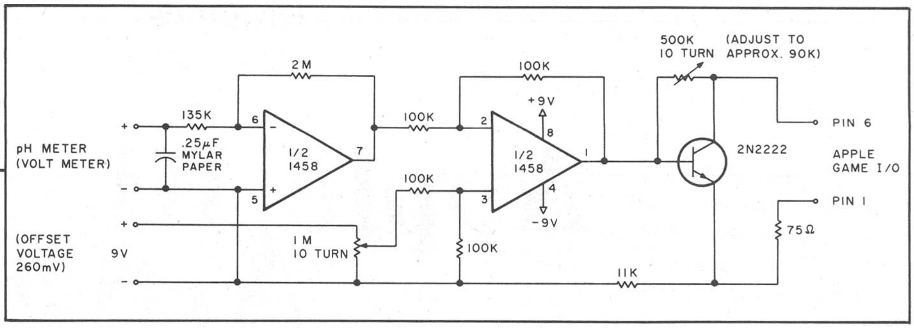
Table 1. Paddle readings for three different pH reference solutions.

Listing continued.

```
 230 LDGT=PH/10:R=0:L=10:N=0
 240 GOSUB (1000+100*LDGT)
 250 REM  NOW DRAW RIGHT DIGIT
 260 RDGT=PH-LDGT*10
 270 R=24:L=0: GOSUB (1000+100*RDGT)
 280 IF LDGT>=7 THEN GOSUB 2000: REM           PLAY A TUNE
 290 FOR DLAY=1 TO 300: NEXT DLAY: REM  DELAY LOOP
 300 COLOR=0
 310 FOR L=21 TO 39: HLIN 0,39 AT L
 320 NEXT L: COLOR=15: GOTO 160
 330 REM  DRAW TEXT PH
 340 VLIN 9,20 AT 11: VLIN 9,20 AT 12
 350 VLIN 9,16 AT 17: VLIN 9,16 AT 18
 360 HLIN 13,16 AT 9: HLIN 13,16 AT 10
 370 HLIN 13,16 AT 15: HLIN 13,16 AT 16
 380 VLIN 3,16 AT 23: VLIN 3,16 AT 24
 390 VLIN 3,16 AT 31: VLIN 3,16 AT 32
 400 HLIN 25,30 AT 9: HLIN 25,30 AT 10
 410 RETURN
1000 REM  DRAW "0"
1010 VLIN 26,39 AT L+R: VLIN 26,39 AT L+R+1
1020 VLIN 26,39 AT L+R+6: VLIN 26,39 AT L+R+7
1030 HLIN L+R,L+R+7 AT 24: HLIN L+R,L+R+7 AT 25
1040 HLIN L+R,L+R+7 AT 39: HLIN L+R,L+R+7 AT 38
1050 RETURN
1100 REM   DRAW "1"
1110 VLIN 24,39 AT L+R+6: VLIN 24,39 AT L+R+7
1120 RETURN
1200 REM DRAW "2"
1210 HLIN L+R,L+R+7 AT 25: HLIN L+R,L+R+7 AT 24
1220 VLIN 24,32 AT L+R+6: VLIN 24,32 AT L+R+7
1230 HLIN L+R,L+R+7 AT 32: HLIN L+R,L+R+7 AT 31
1240 VLIN 32,39 AT L+R: VLIN 32,39 AT L+R+1
1250 HLIN L+R,L+R+7 AT 39: HLIN L+R,L+R+7 AT 38
1260 RETURN
1300 REM  DRAW "3"
1310 HLIN L+R,L+R+7 AT 25: HLIN L+R,L+R+7 AT 24
1320 VLIN 25,39 AT L+R+6: VLIN 25,39 AT L+R+7
1330 HLIN L+R+1,L+R+7 AT 32: HLIN L+R+1,L+R+7 AT 31
1340 HLIN L+R,L+R+7 AT 39: HLIN L+R,L+R+7 AT 38
1350 RETURN
1400 REM DRAW "4"
1410 VLIN 24,39 AT L+R+6: VLIN 24,39 AT L+R+7
```

Listing continued.

Figure 1.
Circuit diagram for circuit that converts a voltage signal into a resistance.



the statements in the program are dedicated to showing the large letters and numbers on the screen.

Demonstration

To demonstrate the system, put some vinegar and water into a beaker and use the pH meter and Apple II setup to display the pH on the CRT. Then neutralize the solution using an Alka-Seltzer tablet. When the pH reaches 7.0, the Apple II will play a little tune (guess what?). To push the pH even higher, add some baking soda.

Conclusion

The small circuit shown is sufficient to demonstrate the simple interfacing of a laboratory instrument to an Apple II microcomputer. It is no precision system, but the results are repeatable and close enough for classroom demonstrations. Total hardware cost of the interface is less than $20. ■

Listing continued.

```
1420 VLIN 24,32 AT L+R: VLIN 24,32 AT L+R+1
1430 HLIN L+R,L+R+7 AT 32: HLIN L+R,L+R+7 AT 31
1440 RETURN
LIST 1500,3150
1500 REM DRAW "5"
1510 HLIN L+R,L+R+7 AT 25: HLIN L+R,L+R+7 AT 24
1520 VLIN 25,32 AT L+R: VLIN 25,32 AT L+R+1
1530 HLIN L+R,L+R+7 AT 32: HLIN L+R,L+R+7 AT 32
1540 VLIN 32,39 AT L+R+6: VLIN 32,39 AT L+R+7
1550 HLIN L+R,L+R+7 AT 39: HLIN L+R,L+R+7 AT 38
1560 RETURN
1600 REM  DRAW "6"
1610 VLIN 24,39 AT L+R: VLIN 24,39 AT L+R+1
1620 HLIN L+R,L+R+7 AT 32: HLIN L+R,L+R+7 AT 31
1630 VLIN 32,39 AT L+R+6: VLIN 32,39 AT L+R+7
1640 HLIN L+R,L+R+7 AT 39: HLIN L+R,L+R+7 AT 38
1650 RETURN
1700 REM  DRAW "7"
1710 HLIN L+R,L+R+7 AT 25: HLIN L+R,L+R+7 AT 24
1720 VLIN 25,39 AT L+R+6: VLIN 25,39 AT L+R+7
1730 RETURN
1800 REM DRAW "8"
1810 HLIN L+R,L+R+7 AT 23: HLIN L+R,L+R+7 AT 24
1820 VLIN 25,39 AT L+R: VLIN 25,39 AT L+R+1
1830 VLIN 25,39 AT L+R+6: VLIN 25,39 AT L+R+7
1840 HLIN L+R,L+R+7 AT 31: HLIN L+R,L+R+7 AT 32
1850 HLIN L+R,L+R+7 AT 39: HLIN L+R,L+R+7 AT 38
1860 RETURN
1900 REM  DRAW "9"
1910 HLIN L+R,L+R+7 AT 24: HLIN L+R,L+R+7 AT 25
1920 VLIN 25,39 AT L+R+6: VLIN 25,39 AT L+R+7
1930 VLIN 25,32 AT L+R: VLIN 25,32 AT L+R+1
1940 HLIN L+R,L+R+7 AT 32: HLIN L+R,L+R+7 AT 31
1950 RETURN
2000 REM PLAY A TUNE USING MUSIC
2010 REM ROUTINES FROM PROGRAMMER'S AID
2020 M=-10473:P=767:T=766:TIMBRE=765
2030 POKE TIMBRE,32:W=120
2040 POKE P,20: POKE T,3*W/4: CALL M
2050 POKE P,15: CALL M
2060 POKE P,22: CALL M
2070 POKE P,15: CALL M
2080 POKE P,24: POKE T,3*W/4: CALL M
2090 POKE P,24: POKE T,W/4: CALL M
2100 POKE P,25: CALL M
2110 POKE P,27: CALL M
2120 POKE P,25: POKE T,W/2: CALL M
2130 POKE P,24: POKE T,W/4: CALL M
2140 POKE P,22: POKE T,3*W/4: CALL M
2150 RETURN
3000 CALL -936: PRINT "      PH METER DISPLAY"
3010 PRINT
3120 PRINT "      PRESS RETURN TO START"
3130 INPUT ST$
3140 GOTO 130
3150 END
```

Apple III Artisan

Put away your Business Basic, and find your smock. . . time to color your III with more than spreadsheets and budgets.

by Bill O'Brien

When my Apple III first walked into the house, I was thrilled. I had a machine with good potential for graphics (no, not great, but better than my 8032 or my TRS-80 Model III). Why, in black and white I could squeeze in 560 horizontal dots and 192 vertical dots. (Hang in there, I'll get to color modes shortly.) I even bought a joystick for all the great things they can do. Since there are two joystick ports on the back of the machine, I couldn't wait to become graphically oriented.

The letdown arrived shortly thereafter. Aside from Desktop Plan III, no one was doing anything with the graphics. Here was this fantastic opportunity to run a game in 60K of program space and everybody's just sitting on their hands!

(Let's take a quick aside: Yes you can run the Apple III in Apple II emulation mode, but the joystick peeks seem to be split between the two ports on the back of the III. Yes, the Apple III was meant to be used as a business machine, but after a hard day of data entry I really could use a few jolts of Galaxian. In fact, this feeling seems common among most of the people for whom I've installed machines. You know there must be some validity to it when, after setting up a $13K multiuser system, the first question you're asked is: "Are there any games for it?")

Well, I couldn't just sit idly by and let such a situation continue. I took out my trusty Apple Business Basic Reference Manual Volume 2, and my equally valuable Owner's Guide, and set out to perform a task.

What resulted wasn't a game (rats!), just a program, or rather the beginnings of a program, that uses the joystick port to draw on the Apple III graphics screen. We'll take a look at it, as well as at some of the available graphics functions. By the time you're done, you'll know enough to take these sparse lines and flesh them out to fit whatever needs you have.

Let's Start Typing

```
10 HOME
1000 ON KBD GOTO 1420
1020 PREFIX$ = ".D1"
```

In the beginning there was the Home command, and when it was given, the chaos of the screen cleared and gave way to calm and the void.

Home, of course, clears the screen and deposits the cursor in the upper left corner. The ON KBD statement alerts the machine to an eventual action. It says, "Keep going with the rest of the program, but if a key is pressed, immediately go to the line number I'm now telling you about."

PREFIX$ is a reserved variable. You can use it in a program, but it has a special meaning. It reminds the Apple III of the route it should take whenever it does any disk input or output. This may not seem like a big deal to you. However, if you have a single drive system, try (if you haven't already) removing the boot disk, putting another disk in its place, and doing *anything* at all with this other disk. The only response you'll get is a VOLUME NOT FOUND error message.

All the storage devices connected to the Apple III have more or less logical ways of being addressed. At most, you can hang four floppy disk drives. By now you may suspect that, as far as the computer is concerned, they're called .D1, .D2, .D3 and .D4. If you have a hard disk connected, you could, for lack of creativity, call it .PROFILE.

What's with all the dots, you might ask? The dot at the start of the drive name tells the Apple that this is actually a route to the device you want to access. For that reason it's called a Pathname. When you first turn on your system the only thing the Apple's really sure of is that it has one disk drive. The default drive is therefore .D1.

When you say SAVE MYPROGRAM.BAS, you're actually instructing the computer to SAVE .D1/MYPROGRAM.BAS. The Ap-

Address correspondence to William O'Brien, WABASA Consulting & Management, 111 Brook St., Scarsdale, NY 10583.

ple is assuming the default prefix to your program. If you said .D2/MY-PROGRAM.BAS, or .D3, or any of the drive names, your program would have been routed to the path you named. Just make certain you really have the item named, or you'll be severely chastised by the computer! To this day I don't know how they got the little hand and the whip in there.

You can change the default drive by assigning its name to PREFIX$ as in line 1020. (In this case, you've just reassured yourselves that drive 1 actually is the default.)

```
1040 OPEN#1,".GRAFIX"
```

Oh no, another dot! Yes, this is also a Pathname, but not to a storage device. The Apple III can access driver routines as well as disk drives. Driver routines are all part of that wonderful file called SOS.DRIVER. They can be used simply by opening them, just as if they were any other file you wanted to handle. You do know what drivers you have in SOS.DRIVER, don't you?

Hang On

At this point you might want to take a deep draught of your favorite beverage. You may get a little nervous. Even if this isn't the case, your throat will probably be grateful for the refreshment.

You should now find a disk that contains these files:

SOS.KERNEL
SOS.DRIVER
SOS.INTERP
BGRAF.INV

If your knees are beginning to shake, you can find them (the files, not your knees) on the Apple Business Basic disk. But there's not much room there to store another program. If you'd like to create your own disk, merely format a blank with the System Utilities disk, and copy these four files from the Basic disk (*not* from the Utilities disk).

The SOS.DRIVER is a collection of files used by the operating system to route all of the input/output devices. The drives are also in there, but Apple was smart. You can't really play with them.

Boot the computer using the Utilities disk and select the System Configuration Program option. (For the snobs in the audience, that's SCP.) When the next menu appears, press the R key for READ DRIVER. If you have only the one disk drive, remove the Utilities disk and replace it with a disk you've previously used to boot the machine (the Basic disk would be the perfect choice). If you have two drives, insert the other disk in drive 2.

If you have one drive, just press return. If you have two, change the .D1/SOS.DRIVER to .D2/SOS.DRIVER. Press return.

Whirr. Click. . . .

Sometime in the near future the screen will partially clear and a list will appear that might look something like this:

1. CONSOLE
2. AUDIO
3. PRINTER
4. SILENTYPE
5. PROFILE
6. GRAFIX
7. PARALLEL
8. RS232

Take Your Pick

How many of these you have, or what order they're in, isn't really important. Since all of the above files can be used, and all take up memory when they're loaded in, you must choose between being always prepared and having the most free memory. Let's look at this list.

Console—That's necessary, for without it you wouldn't be able to see anything on the screen.

Audio—If you want to turn your Apple III into an electronic organ, fine. We don't need it now.

Printer—As supplied by Apple, it's for a Qume printer. Or you may have altered it to accommodate some other printer. If you want to dump a listing of this program to your printer, then retain this specification.

Silentype—Only for the Apple Silentype printer. If you don't have one, leave it off. Same rule applies here as for Printer. If it's your listing device, keep it; if not, give it the axe.

Profile—This is the hard disk driver. If you don't have one, the odds are good you won't have this driver. If you have a hard disk drive, retain this driver.

Grafix—I think we'll hang on to this one for now.

Parallel—This is another "option" driver that might be on your system as the driver for a parallel printer or for parallel communications with another machine.

RS232—This is a bidirectional serial driver, usually used to communicate over the phone lines with a modem. Unless you're using it as a printer driver, you don't really need it.

If you happen *not* to find Grafix, get a disk called System Utilities Data. Use the Read option of the SCP. . . there, now we've all become snobs. Read in Grafix.Driver, using either a .D1/ or .D2/, as your system requires.

To disable any of the drivers after you've read them in, press the escape key and choose Edit. Yes, you can remove them completely, but why go through the bother of restoring them later? When the drivers are listed, press the number for the one you want to omit, and then press the return key. For the next choice select 4, Status, and make the driver inactive. This will keep it on the disk, but will not load it when you boot. Repeat these steps for all drivers you don't need.

Help

If you run into problems, or are a bit squeamish about getting into this level of modification just now, you have two choices. Either spend a quiet evening alone with the Device Drivers Manual and practice on a lot of *copies* of disks, or go back to your dealer and boldly say, "I'd like a little support, please."

There shouldn't be any problems. (Just one favor—don't call me at 8:30 on a Sunday morning. Yes, I know you're in England and it's 1:30 in the afternoon there, but here in the States Saturdays are very long evenings that sometimes chase Sunday morning.)

Since all driver files can be accessed like regular files, we'll use line 1040 to open one up.

Witchcraft?

```
1060 INVOKE"BGRAF.INV"
1070 PERFORM INITGRAFIX
```

Line 1060 comes from a musty old tomb found in the ruins of a burned-out cottage in Salem. It releases the screen demons.

Actually, the BGRAF.INV file that

0	Black
1	Magenta
2	Dark Blue
3	Purple
4	Dark Green
5	Grey 1
6	Medium Blue
7	Light Blue
8	Brown
9	Orange
10	Grey 2
11	Pink
12	Green
13	Yellow
14	Aqua
15	Black

Table 1.
The Apple III colors.

Mode Numbers	Memory Used	Description
0	8K	280 × 192 black & white
1	16K	280 × 192 limited color
2	16K	560 × 192 black & white
3	16K	140 × 192 full color

Table 2. The four available graphics modes.

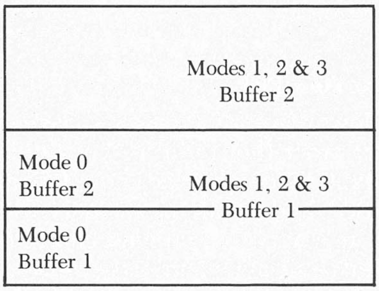


Figure 1. Memory management of modes and buffers.

we made sure we had along with the SOS routines, is a machine-language subroutine. By invoking this file, you load it into memory and ready it for use. Good as our word, line 1070 makes immediate use of it.

When you "perform" an action related to an invokable file, you send the little Apple III messenger running through the RAM stacks looking for that section of the file you want to perform. In this case, its called INITGRAFIX. As part of BGRAF.INV, it shuffles your program around in memory so it won't be destroyed when you use other graphic routines. Essentially, it initializes the graphic environment.

Adding Color

```
1100 PCOLOR = 15
1120 BUFFER = 1
1140 MODE = 2
1160 PERFORM GRAFIXMODE (%MODE,
%BUFFER)
1180 PERFORM GRAFIXON
1200 PERFORM PENCOLOR (%PCOLOR)
1220 COLOR% = 0
1240 PERFORM FILLCOLOR
(%COLOR%)
1260 PERFORM FILLPORT
```

Could someone please assist those 14 people who have just fainted?

Lines 1100–1140 create variables you'll use later on. They really could've been called anything, but I chose PCOLOR because that's the value the Pen COLOR will be, BUFFER because you'll be using a graphic "buffer," and MODE because you'll also be using one of the graphic "modes." I like to keep everything orderly, and since it's my program, to quote Mel Brooks, "It's good to be the King."

All of the drawing on an Apple III screen is done with a pen. And pens, of course, come in different colors; Table 1 shows what's available. The four modes are shown in Table 2. For each of the modes, two buffers, or drawing areas, can be used. That means you'll need a maximum of 32K of memory for drawing.

Mode 0 is straightforward. Beginning in the lower left corner, which is the Home, or 0,0 position, for the graphic screen, you can handle 280 dots across (0 to 279) and 192 dots from bottom to top (0 to 191).

Mode 1 is a little tricky. You have the same number of horizontal and vertical positions, but in color. Because it's a color mode, it imposes a few restrictions. The 280 horizontal dots are actually composed of 40 groups of seven dots each. When you turn on (paint) any of the dots in any "family," the painted dots in that group of seven will all be the same color.

If, at some later time, you change the pen color and paint one of the dots in the group, all of the turned-on dots in that same group will change to the new color. (The dots that are on are said to be in the foreground, while the dots that are off are in the background. The same rule holds true for the background dots. Changing the color of any one dot in the background will change the color for all background

LET US ENTERTAIN YOU!

Avant-garde means "ahead of its time" and Avant-Garde Creations truly lives up to its name with exciting entertainment software that promises more than the mundane! Not just the usual bang-bang shoot-em-up types, but challenging games that require skill, timing, fast thinking and finesse.

ZERO GRAVITY PINBALL keeps you busy even at speed 1 with 10 flippers, 2 force fields, an unpredictable black hole and of course zero gravity.

LAZER MAZE demands a keen eye and a quick trigger finger. The mirror maze challenges perceptual skills you probably didn't even know you had!

FEDERATION puts you in command of a space fleet on a mission to wipe out the merciless invading Drorn Drones. Its lightening fast action requires perfect timing.

DEATH RACE '82 combines the skill of high-speed driving with the thrill of a perilous chase as you attempt to escape from robot guards in Death Squad Cars.

HI-RES COMPUTER GOLF, our all-time best seller, is better than the real thing! This highly acclaimed game requires strategy, judgement, visual keeness and skill to meet the challenge of five 18-hole courses. Yelling "Fore!" is optional.

All of these great games run on Apple II Plus 48K and retail for $29.95 They are available at your local dealer or you may order direct from Avant-Garde Creations.



AVANT-GARDE CREATIONS
P.O. BOX 30160 • EUGENE, OR 97403 • (503) 345-3043
BUSINESS, EDUCATION, UTILITY AND ENTERTAINMENT SOFTWARE

Circle 52 on Reader Service card.

dots of the same group.)

Mode 2 is absolute ecstasy. As well as the 192 (0–191) vertical dots, you have 560 of those little buggers to play with (0 to 559) horizontally; the amount of medium detail drawing is excellent.

Mode 3 is similar to Mode 1, but each of the 140 horizontal or 192 vertical dots can be painted independently of the others.

Buffers are the areas in memory used to hold the data you want to display on the screen. With a possible 116K of memory to be used (three 16K modes, with two buffers each, and one 8K mode, also with two buffers), you might assume that the different modes and buffers share memory space—they do. In fact, as I said, we only need 32K of buffer to move around in.

Since you can switch back and forth between buffers, you can have a picture in Mode 0, Buffer 1, in Mode 0, Buffer 2 *and* in Mode 1 or 2 or 3, Buffer 2. By changing the values assigned to the variables MODE and BUFFER and issuing a PERFORM GRAFIXMODE(%MODE,%BUFFER) command, you can switch back and forth between these three areas.

You cannot use Mode 0 at all, though, if you're using Buffer 1 of any of the other modes, since they overwrite the Mode 0 areas. Nor can you use the same buffer for any of the other modes, since they each share the same 16K buffer areas. See Figure 1. (Switching from Mode 1, Buffer 1 to Mode 2, Buffer 1 will distort the contents, since the display options do not coincide. Switching to Mode 2, Buffer 2, however, lets you jump back and forth between the two screens.)

Line 1160 performs a subroutine (just like INITGRAFIX), but this time you're passing two variables into it, telling the graphic system which mode and which buffer of that mode to use (in this case, Mode 2 and Buffer 1—but you knew that from lines 1120 and 1140, right?)

The % signs are small tranquilizers that you *must* use to reassure the graphic system that the variables we are passing are integers. Had you defined MODE as MODE%, you would still have to use the leading % symbol.

Line 1180 turns on the graphic screen. You're getting used to this, aren't you? Line 1200 sets the pen color (if you guessed it was white, go to the head of the class and keep reading). Another variable is created in 1220. This is the color used to fill in the background every time you clear the screen. Set that color in line 1240. Just as in normal screen work, "clearing the screen" is simply filling the screen with some color. Line 1260 performs the filling (notice that even though you defined COLOR% as an integer, you still had to use the leading %).

Digest all of that for a moment while I recap for the West Coast.

To do graphics, you use a machine-language module called BGRAF.INV, which is called (invoked) into memory. Within BGRAF, subroutines handle (PERFORM) the tedious work of selecting and setting graphic points on the screen.

Onward!

```
1280 FLAG = 1
1300 Y% = PDL(3)*.74902:
X% = PDL(2)*2.19216
1310 GOTO 5700
```

The variable in line 1280 will be used later on to remind the program that we've already been through here. Again, it could have been called anything, but since it's a flag, the name FLAG seems appropriate.

Rattle Your Paddle

The two joystick ports on the back of the Apple III can each read two paddle inputs. Typically these are variable resistors that range from 0 to 150k ohms. I'll be working with port A. If you have a Silentype printer, use port B and change line 1300 to read

```
1300 Y% = PDL(1)*.74902:X% = PDL(0)*
2.19216
```

Line 1300 reads the Y (vertical) axis of the screen from PDL(3), while the X (horizontal) axis is obtained from PDL(2). The only problem is that the input from PDL will be in the range of 0 to 255 in each direction. You'll have to scale these figures to work within the selected mode. I've chosen Mode 2, so I scale:

560/255 = 2.19216 for the horizontal
192/255 = .74902 for the vertical

In any of the other modes, you would substitute 280 or 140 for the 560 to get the correct horizontal factor. In case

THE PEACH™ WITHOUT FUZZ



The Peach™ is the newest addition to Electronic Protection Devices crop of EMI/RFI Filters/AC Surge Protectors. It eliminates transients such as "spikes" or "glitches" same as The Lemon™, The Lime™ and The Orange™ while simultaneously filtering out "fuzz" or "noise" produced by Elecro Magnetic Interference (EMI) or Radio Frequency Interference (RFI).

The increasing complexity and scope of modern electronics demands that each microprocessor controlled product perform its function without extraneous signals of any kind that would degrade or reduce the intelligibility of that product. "Hash"- electrical noise from rapid opening and closing of contacts or "glitches" - random noise pulses that produce small disturbances in the baseline of your CRT display interfere with normal operation by causing errors in data transmission. Data errors can lead to skewed results, lost time and aggravation.

Prevent this from happening to you with The Peach. Each Peach is a solid state clamping device with EMI/RFI filtering utilizing high speed semi-conductor technology. Simply plug The Peach into any standard 3 wire duplex outlet then plug what needs protection into it. Each Peach has 3 outlets and exceeds the IEEE 587-1980 Guide for Surge Voltages in Low Voltage Power Circuits.

Compare the cost of computer hard-ware, software and your time with the price of our Peach ($97.50). You'll opt for a line free from surges and no fuzz with The Peach from EPD. Available through your local dealer.

EPD
Electronic Protection Devices
5 Central Avenue
Waltham, Massachusetts 02154
(617) 891-6602
1-800-343-1813

Circle 158 on Reader Service card.

Key	ASCII Value	Function
C	67	Clear screen
D	68	Draw with dots
E	69	End program
G	71	Save the image
P	80	Draw with lines (paint)
T	84	Add text

Table 3.
ASCII values of function keys.

you hadn't noticed, all modes have the same vertical resolution: 192.

As it stands, line 1300 takes care of all that. If you do enlarge on the program, a few well chosen If. . . Then statements could handle the factor options. I'll get to line 5700 a little later in this broadcast.

Pushing Its Buttons

```
1420 IF KBD = 67 THEN PERFORM FILL-
     COLOR (%COLOR%): PERFORM
     FILLPORT
1425 IF KBD = 71 THEN PERFORM
     GSAVE."PIC" + STR$(COUNT):
     COUNT = COUNT + 1
1430 IF KBD = 69 THEN 1520
1435 IF KBD = 68 THEN 5800
1440 IF KBD = 84 THEN 5000
1445 IF KBD = 80 THEN 5200
1460 ON KBD GOTO 1420
1480 RETURN
```

Remember, way back at the beginning of the program, we specified that if any key were pressed the computer should stop what it was doing and immediately go to line 1420? Well, we're here. These lines help us determine what key was pressed and what should be done because of it.

The variable, KBD, is an integer that contains the ASCII value of the key that was pressed. Table 3 contains a list of the values used. If you'd like to experiment a little, write a one-line program:

```
= => 10 GET A$:PRINT A$;" ";ASC(A$):
GOTO 10
```

To exit from it you'll have to press the reset key, but not the control key. That will give you the ASCII values for every character your keyboard can generate. The only warning here is that the graphics program looks for capital letters; lowercase characters will not work.

Line 1425 deserves a brief note. If, for some egotistical reason, you should happen to like the picture you've created, it would make no sense to destroy it when you had to leave. GSAVE.filename saves the picture (the . must be included). In our case, the variable COUNT is turned into a string and made part of the name so that the first picture saved will be PIC0, the next PIC1, and so on. Take heed! The variable is reset to zero every time the program runs.

```
1500 REM ____________________
1510 REM
1520 FOR X = 1 TO 4:PERFORM RELEASE:
     NEXT X
1540 CLOSE
1560 HOME:PRINT CHR$(15)
1580 END
```

If the program detects an E, it jumps straight to line 1520. It may look like we're freeing prisoners—that's because we are.

Up until now, the buffer memory has been reserved solely for graphic use. Since the program ended when it branched here, it would be nice to have all that memory back. You never know, you might want to use it for writing a program.

PERFORM RELEASE takes care of this matter, and petitions the Apple III memory governor to have all that penned-up memory paroled back into your custody. First it tackles Buffer 2 of Modes 1, 2 and 3, releasing that 16K group; then it moves down to Buffer 1 of those same modes and does the same. Buffer 2, Mode 0 follows, joined shortly thereafter by Buffer 1. If you haven't used that much memory, it's no great strain to do this four times, since nothing else is affected.

Home, of course, clears the text screen, while printing CHR$(15) switches back to the original type of text display (80-column, 40-column, colored backgrounds). You could have done the same thing by issuing a Text command, but that would reset the text screen to 80-column mode.

Even though this subroutine ends the program, you're not done yet.

```
5000 REM ____________________
5005 REM
5010 PERFORM PENCOLOR(%0):PER-
     FORM DOTAT(%X%,%Y%): PER-
     FORM PENCOLOR(%PCOLOR)
5020 GET A$
5030 IF A$ = CHR$(7) THEN 1460
5040 IF ASC(A$)<32 THEN 5020
5050 PRINT#1;A$;
5060 GOTO 5020
```

More Possibilities

Up until this point (and you'll have to take my word for this until we get to line 5700), we've been moving around the screen having a great time filling in dots and lines and things. We pressed

"...just one favor. Don't call me at 8:30 on a Sunday morning."

the T key, and now the program has changed the pen color to black, put a black dot at the last known X, Y joystick position, returned the pen to its original color and is waiting for us to type in some text. And all of that in only lines 5010–5020!

The two things it looks for are the control-G, which signals the end of text input (line 5030), and any key, or key sequence, that produces a character with an ASCII value less than 32 (line 5040). If it finds the former, it returns to line 1460, where you reinitialize the ON KBD statement and then return to the same point in the program from which ON KBD sent you packing. (And you thought I wasn't going to explain why I used a Return when I hadn't used a Gosub!)

Any other character is printed on the screen using the resident character set and color of whichever mode you're in. The cursor automatically advances one character position and goes back and waits some more.

```
5200 REM______________________
5202 REM
5203 PERFORM PENCOLOR(%PCOLOR)
5205 IF BUTTON(2) < >0 THEN 1460
5210 Y% =PDL(3)*.74902:X% =PDL(2)*
     2.19216
5220 PERFORM LINETO(%X%,%Y%)
5230 GOTO 5205
```

This is the painting portion of the program, where you draw with lines instead of dots (it's a smoother, more continuous line). As long as the joystick button's value is equal to zero (not pressed), line 5220 will continue to place (Lineto) a continuous line of dots from the last position to the new cursor position. Press the button on the joystick, and you return.

```
5600 REM ______________________
5602 REM
5603 PERFORM PENCOLOR(%COLOR)
5605 IF BUTTON(2)< >0 THEN 1460
5610 Y% =PDL(3)*.74902:X% =PDL(2)*
     2.19216
5660 PERFORM DOTAT (%X%,%Y%)
5670 GOTO 5605
```

This is similar to the line drawing routine, except that instead of a continuous line extending from the last cursor position to the next, dots form the path. The end result is a dot pattern that, more often than not, is segmented. This is because the time the program spends checking itself, and then going to the joystick port, is longer than the time it takes you to move the joystick.

```
5700 REM ______________________
5705 REM
5710 :
5715 PERFORM MOVETO (%X%,%Y%)
5720 DCOLR% =EXFN%.XYCOLOR
5730 :
5735 PERFORM PENCOLOR (%0) :PER-
     FORM DOTAT (%X%,%Y%)
5750 PERFORM PENCOLOR (%15) :PER-
     FORM DOTAT (%X%,%Y%)
5770 PERFORM PENCOLOR (%DCOLR%)
     :PERFORM DOTAT (%X%,%Y%)
5780 :
5790 Y% =PDL(3)*.74902:X% =PDL
     (2)*2.19216
5795 GOTO 5700
```

Finally! The fabled line 5700!

This short section is where the program spends most of its time. While it's waiting for a key depression to tell it what specific functions you want done, it polls the joystick, keeping track of its present location. All along, whenever you've seen the joystick's screen location, it's been represented as a flickering dot. The tricky part is to not erase any dot that is already turned on when you flicker past it.

To do that, you need another of the procedures found in BGRAF. This one is called EXFN%.XYCOLOR. I'm betting that by now you know what this does. Just in case you don't, I'll tell you. EXFN%.XYCOLOR reads the color of the dot at the current cursor location. Remember, whether the dot is in the background or the foreground, it will always have a color.

I've assigned the variable DCOLR% (the dot color) to this number. Lines 5735 to 5770 take the current cursor location, put a black dot there, put a white dot there (to get the flicker) and then replace it with a dot of the original color. Then they scan the joystick to see if it has moved, and repeat the process. Without this routine you'd erase whatever was under the dot before you got there.

And that's the entire program.

You're On

What comes to mind immediately is adding a little simple geometry and perhaps making the program recognize the S key to generate shapes. There's also the F key to fill in the shapes. After that, where do you stop? An M function to change buffers, modes and colors?

Why are you sitting around? Don't say I never gave you anything. However, I expect something in return. Somebody, anybody, write just one Apple III game. Get the game published. I'll settle for chess, but please, no Wumpus! ■

Profit, Please

Most businesses prefer profit to loss. This easy-to-use program reveals a company's break-even point. If you don't get the hint, it's your loss!

by Corey D. Schou and Ronald S. Rubin

How can a small business determine if it's turning a profit or losing money? How long will it take for revenues to offset costs? How many products or services must be sold to break even? These questions can be answered through break-even analysis, a simple yet powerful procedure for small business planning.

The technique studies the relationships between sales, fixed costs and variable costs. In general, the break-even point in business occurs when revenues and expenses are equal; the business is neither making a profit, nor experiencing a loss. Break-even analysis is most useful when applied to future periods as a planning guide, particularly if a change in business operation is anticipated. For example, break-even analysis can provide flexible income and expense projections under assumed conditions and alternative managerial programs.

Some Basic Applications

1. New Product decisions: Each new product has cost implications, which obviously affect price and marketability of a new product. In many small businesses, a new product with uncertain sales volume raises the profit potential, as well as the risks, for that company. Break-even analysis permits the comparison of possible cost structures as trade-offs.

2. Expansion decisions: It may be used to study the aggregate effect of a general business expansion. For example, the relationships between total dollar sales and total dollar costs are examined in order to identify potential changes in these relationships.

3. Profit margin decisions: Break-even analysis can be used to determine the sales dollars required to earn a given level of profit.

4. Advertising decisions: It may be used to forecast the sales results needed for an advertising campaign to cover costs.

5. Planning decisions: Use of break-even analysis could enable a business to know if it is on schedule, as far as the overall plan is concerned. There is probably no business that has a nice, even flow of income or expenses, but break-even points provide a ready reference that can be used by the small business to think more about the total plan.

For example, let's assume that the daily break-even point for a particular business is computed to be $400 for each day of operation. A simple mental calculation is all that's required when the day's receipts are added up to determine if the business incurred a loss, broke even or made a profit. A day is normally too short a period of time to make any immediate changes in a business plan, but it would be a simple matter to extend the calculations to a week, two weeks or a month.

Assumptions

This analytical tool makes two assumptions. The first, seemingly reasonable assumption, is that costs can be reasonably divided into fixed and variable components. By definition, fixed costs are those which do not change with the sales volume, e.g., salaries, rental expenses, depreciation expenses, property taxes, most insurance premiums, and so forth.

On the other hand, variable expenses are those which change with the sales volume of the business. At least one type of variable changes directly and proportionally with sales—commission expenses. However, most variable expenses do *not* vary directly and proportionally with sales. In fact, there are degrees of variability. As a result, most variable expenses are actually semi-variable. That is, some costs remain fixed up to a certain sales volume, and then jump as that volume is exceeded.

For example, office costs or delivery expenses may fit into such a category. The key is good judgement in propor-

Address correspondence to Drs. Ronald S. Rubin and Corey D. Schou, Department of Marketing, University of Central Florida, Orlando, FL 32816.

Program listing. Applesoft Basic program for break-even analysis.

```
1  REM    PROGRAM WITH REMARKS IS 3585 BYTES LONG ($0E01) OR 2816 BYTES W
   ITHOUT REMARKS
2  REM ******* VAIRABLE MAP ***********
3  REM  A$=GENERIC STRING INPUT
4  REM  B$ CONTAINS "BREAK EVEN POINT
5  REM  CO$ CONTAINS "CONTRIBUTION MARGIN
6  REM  C$ IS ARRAY WITH LABLES FOR INSTRUCTION SECTION
7  REM  F$ CONTAINS "FIXED COSTS
8  REM  V$ CONTAINS "VARIABLE COSTS
9  REM  DB=DAILY BREAK EVEN POINT
10  REM  BE=BREAK EVEN POINT (YEAR)
11  REM  FC=FIXED COSTS PER DAY
12  REM  PS=PROJECT SALES
13  REM  TF=  TOTAL FIXED COSTS
14  REM  TV=TOTAL VARIABLE COST
15  REM  THIS  IS A PROGRAM TO TEACH ONE ABOUT BREAKEVEN ANALYSIS.
16  HOME
17  HTAB 10: PRINT "BREAK EVEN ANALYSIS"
18  HTAB 18: PRINT "BY"
19  HTAB 11: PRINT "COREY SCHOU, PH.D."
20  HTAB 11: PRINT "RONALD RUBIN, PH.D."
21  VTAB 7: PRINT "THIS PROGRAM DISCUSSES A SIMPLIFIED"
22  PRINT "METHOD OF CALCULATING THE BREAK-EVEN "
23  PRINT "POINT FOR A SMALL RETAIL OPERATION."
24  PRINT "BREAK-EVEN ANALYSIS IS AN EXCELLENT"
25  PRINT "STARTING POINT FOR FINDING OUT WHERE"
26  PRINT "A SMALL BUSINESS IS, AND MORE "
27  PRINT "IMPORTANTLY, WHERE IT CAN GO."
28  PRINT "IT PROVIDES A FINANCIAL PLANNING"
29  PRINT "TAKEOFF POINT FOR SMALL STORES."
30  GOSUB 139: REM  ENDS COVER PAGE
31  HOME : HTAB 10: PRINT " WOULD YOU LIKE:"
32  VTAB 3: HTAB 7: PRINT "1. DEMONSTRATION OF TECHNIQUE"
33  HTAB 7: PRINT "2. BREAK-EVEN CALCULATIONS"
34  VTAB 24: HTAB 10: PRINT "TYPE 1 OR 2 ONLY";
35  GET A$: IF A$ = "1" THEN 39
36  IF A$ = "2" THEN 96
37  GOTO 31
38  REM  END INDEX
39  HOME : REM  BEGIN INSTRUCTION PORTION
40  LET F$ = "FIXED COSTS":V$ = "VARIABLE COSTS":B$ = "BREAK EVEN POINT":
    CO$ = "CONTRIBUTION MARGIN"
41  DIM C$(10,2,2)
42  FOR Z = 1 TO 2: FOR X = 1 TO 7: FOR Y = 1 TO 2
43  READ C$(X,Y,Z)
44  NEXT Y,X,Z
45  INVERSE : PRINT "PROJECTED SALES = $51,000"
46  PRINT
47  FOR Z = 1 TO 2
48  INVERSE
49  IF Z = 1 THEN  PRINT F$
50  IF Z = 2 THEN  PRINT V$
51  NORMAL
52  FOR X = 1 TO 7
53  LET T =  LEN (C$(X,1,Z)):T1 =  LEN (C$(X,2,Z))
54  PRINT C$(X,1,Z); TAB( 39 - T1 + T2);C$(X,2,Z)
55  NEXT X
56  PRINT
57  NEXT Z
58  GOSUB 139
59  VTAB 20
60  HTAB 1
61  PRINT "              SALES=100.00%"
62  PRINT "      VARIABLE COSTS= 60.12%";
63  FLASH
64  PRINT "(30665/51000)"
65  VTAB 22
66  NORMAL
67  PRINT CO$;"= 39.88%"
68  VTAB 1: HTAB 20: FLASH : PRINT "51,000": NORMAL
69  VTAB 19: HTAB 34: FLASH : PRINT "30665": NORMAL
70  INVERSE
71  GOSUB 139: VTAB 23: HTAB 1: PRINT "NOW TO CALCULATE THE  BREAK-EVEN P
    OINT"
72  NORMAL
73  GOSUB 139
74  HOME
75  LET T =  LEN (B$):T1 =  LEN (F$):T2 =  LEN (CO$)
76  VTAB 10
77  PRINT B$;"=";
78  FOR X = 1 TO 37 - T
79  PRINT "_";
80  NEXT X
```

Listing continued.

tioning various costs between fixed and variable categories.

The second assumption is that all costs-volume-profit relationships are linear. But doubling of sales volume may, or may not, double the revenue, because a reduction in the unit price may have been necessary to achieve the volume increase. However, for the limited range surrounding the break-even point, assume it is accurate enough to be useful. For example, to examine the effect of doubling the level of operations, you would have to develop new (linear or nonlinear) relationships.

The Break-even Technique

The basic formula is:

$$\text{Revenue} = \text{Fixed Costs} + \text{Variable Costs}$$

The difference between revenue and variable cost is called the Contribution Margin (CM). This represents each unit's contribution toward covering the fixed costs and thereby making a profit.

$$\text{CM} = \text{Revenue} - \text{Variable Costs}$$

If CM is expressed as a percent of revenue, this is found by setting sales equal to 100 percent and finding the percentage of variable costs in relation to sales. Subtracting that figure gives you the Contribution Margin.

For example, if sales were projected to be $51,000, and variable costs are $30,665, then the variable costs represent 60.12 percent of the 100 percent (representing sales). We obtain a Contribution Margin of 39.88 percent. The break-even level is calculated by dividing the fixed costs by CM. (Remember, CM is expressed as a percentage of revenue).

$$\text{Break-even Point (in \$)} = \frac{\text{Fixed Costs (in \$)}}{\text{CM (\% of Revenue)}}$$

Case Example

Consider this situation. A beauty shop owner is considering operation expansion. The owner has identified the fixed and variable costs as presented in Table 1.

The owner estimates that sales could be $51,000 for the expansion year. He wants to know what the break-even point is for that particular year, and how much revenue must be obtained per day in order to break even. The an-

More Apple II owners choose Hayes Micromodem II than any other modem in the world. Compare these features before you buy. You should. It's your money. Thousands of other Apple II owners have already compared, considered, and are now communicating — all over the U.S.A. — with Micromodem II. The best modem for the Apple II. The most modem for your money.



A complete data communication system. Micromodem II is not "base priced" plus necessary "options." It's a complete, high-performance data communication system. The printed circuit board fits — quickly and easily — into your Apple II, eliminating the need for a serial interface card. And the Microcoupler™ (included) connects the Apple II directly to a standard modular telephone jack. Auto-dial and -answer features are built in. Operation can be full or half duplex, with a transmission rate of 300 bps. And it's Bell 103 compatible and FCC approved.

Now there's Hayes Terminal Program, too! Developed by Hayes specifically for Micromodem II, this new Terminal Program allows you to access all the great features of your modem in a matter of seconds.

With it, you can use your CP/M,® DOS 3.3 or Pascal formatted disks to create, send, receive, list and delete files. Hayes Terminal Program is a complete, stand-alone disk.

And because it's menu driven, you can choose from a wide variety of options to set your communication parameters — as well as change hardware configuration — directly from the keyboard. It even allows you to generate ASCII characters that are normally not available from Apple keyboards, further extending your capabilities. Incoming data can be printed (on serial or parallel printers) as it's displayed on your screen.

Micromodem II is available with or without the Terminal Program. Buy your modem by itself, or optionally packaged with the Terminal Program disk and user manual at extra cost. The software is also sold separately, for those who already own a Micromodem II.

If you're ready to communicate with other computers, to access information utilities, time-sharing systems, or use bulletin boards, then you're ready for Micromodem II. Come on. Compare. Consider. Then buy.

Micromodem II is already the best-selling modem for the Apple II. And Hayes' new Terminal Program makes it better than ever. Available at computer stores nationwide.

Hayes

Hayes Microcomputer Products, Inc.
5835 Peachtree Corners East
Norcross, Georgia 30092 (404) 449-8791



Circle 51 on Reader Service card.

Circle 2 on Reader Service card.

Inside Apple

Apple Computer Inc., 20525 Mariani Avenue, Cupertino, California 95014 — Vol. 1 No. 2

For the authorized Apple dealer nearest you, call 800-538-9696 (800-662-9238 in California.)

Fruitful Connections.

There are more people in more places making more accessories and peripherals for Apples than for any other personal computer in the world.

Thanks to those people — in hundreds of independent companies — you can make the humblest 1978 Apple II turn tricks that are still on IBM's Wish List for 1984.

But now we're coming out with our very own line of peripherals and accessories for Apple® Personal Computers.

For two very good reasons.

First, compatibility. We've created a totally kluge-free family of products designed to take full advantage of all the advantages built into every Apple.

Second, service and support.



Now the same kindly dealer who keeps your Apple PC in the pink can do the same competent job for your Apple hard-disk and your Apple daisywheel printer.

So if you're looking to expand the capabilities of your Apple II or III, remember:

Now you can add Apples to Apples.

A joy to behold.

The new Apple Joystick II is the ultimate hand control device for the Apple II.

Why is it such a joy to use?

With two firing buttons, it's the first ambidextrous joystick — just as comfortable for lefties as righties.

Of course, it gives you 360° cursor control (not just 8-way like some game-oriented devices) and full X/Y coordinate control.

And the Joystick II contains high-quality components and switches tested to over 1,000,000 life cycles.

Which makes it a thing of beauty. And a joystick forever.



Gutenberg would be proud.

Old Faithful Silentype® has now been joined by New Faithfuls, the Apple Dot Matrix Printer and the Apple Letter Quality Printer.



So now, whatever your budget and your needs, you can hook your Apple to a printer that's specifically designed to take advantage of all the features built into your Apple. With no compromises.

The 7x9 Apple Dot Matrix Printer is redefining "correspondence quality" with exceptional legibility. With 144x160 dots per square inch, it can also create high resolution graphics.

The Apple Letter Quality Printer, which gets the words out about 33% faster than other daisywheel printers in its price range, also offers graphics capabilities. See your authorized Apple dealer for more information and demonstrations. Because, unfortunately, all the news fit to print simply doesn't fit.



A storehouse of knowledge.

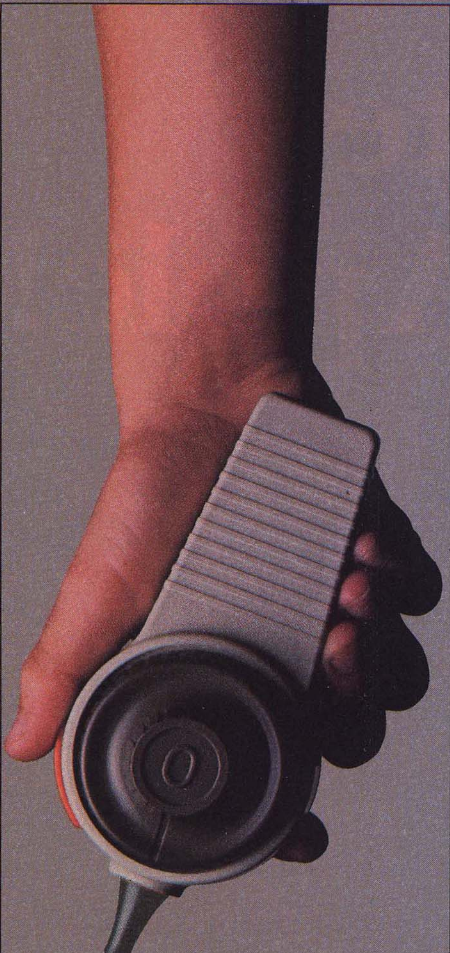
If you work with so much data or so many programs that you find yourself shuffling diskettes constantly, you should take a look at Apple's ProFile,™ the personal mass storage system for the Apple III Personal Computer.

This Winchester-based 5-megabyte hard disk can handle as much data as 35 floppies. Even more important for some, it can access that data about 10-times faster than a standard floppy drive.

So now your Apple III can handle jobs once reserved for computers costing thousands more.

As for quality and reliability, you need only store one word of wisdom:

Apple.



Up the creek without a paddle?

Or lost in space? Or down in the dungeons?

Whatever your games, you'll be happy to know that someone has finally come out with game paddles built to hold up under blistering fire. Without giving you blisters.

Apple Hand Controller II game paddles were designed with one recent discovery in mind:

People playing games get excited and can squeeze very, very hard.

So we made the cases extra rugged. We used switches tested to 3,000,000 life cycles. We shaped them for holding hands and placed the firing button on the right rear side for maximum comfort.

So you'll never miss a shot.

Launching pad for numeric data.

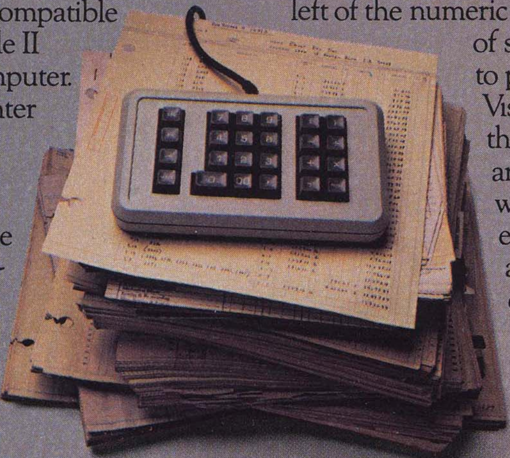
Good tidings for crunchers of numerous numbers:

Apple now offers a numeric keypad that's electronically and aesthetically compatible with the Apple II Personal Computer. So you can enter numeric data faster than ever before.

The Apple Numeric Keypad II has a standard calculator-style layout. Appropriate, because unlike some other keypads, it can actually function as a calculator.

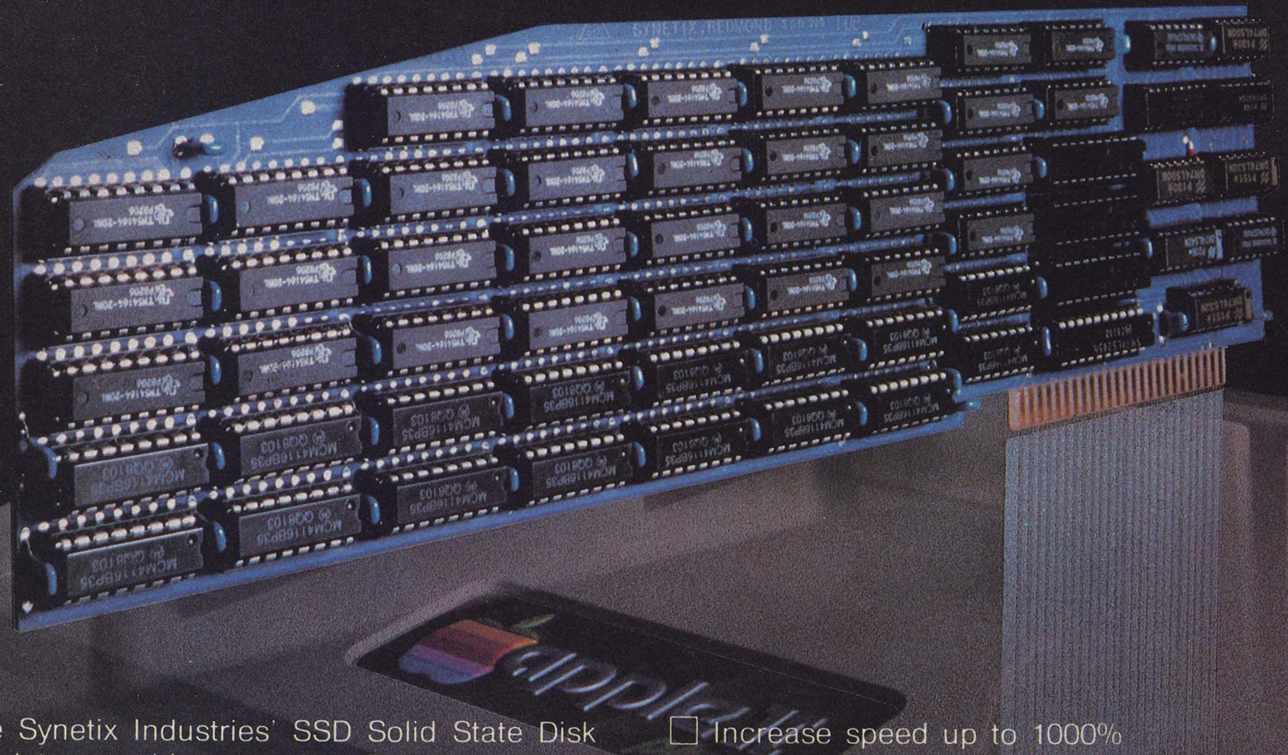
The four function keys to the left of the numeric pad should be of special interest to people who use VisiCalc.® Because they let you zip around your work sheet more easily than ever, adding and deleting entries.

With one hand tied behind your back.



Circle 10 on Reader Service card.

294 K Bytes Out of your Apple!



The Synetix Industries' SSD Solid State Disk Emulator provides complete Plug-In emulation of either a Single (147 K Byte) or Dual (294 K Byte) Disk Drive. The memory board will operate in any slot 1 through 7. Total software compatability is maintained for *APPLE DOS 3.3, APPLE PASCAL *CP/M. Up to seven SSD's may be used depending on other peripheral equipment in use.

- ☐ Software compatible with *APPLE DOS 3.3, *PASCAL, & *CP/M operating systems
- ☐ Automatic copying of Disks into SSD Memory
- ☐ Single Drive 147 K Byte
- ☐ Dual Drive 294 K Byte
- ☐ Increase speed up to 1000%
- ☐ Add up to 2 Megabytes of Solid State Memory to the APPLE
- ☐ Operates in any I/O slot 1 through 7
- ☐ Increase reliability by reducing mechanical disk drive failures
- ☐ Requires no external power or modifications to the APPLE
- ☐ Price competitive with mechanical drives
- ☐ Reduces space requirements

Micro Computer Product Marketing
Synetix Industries, Inc.
15050 NE 95th St., Redmond, WA 98052
206-885-4215

Creative Products by Synetix

Call Toll Free 1-800-426-7412

Listing continued.

```
 81  PRINT " "
 82  VTAB 9: HTAB (39 - T): PRINT F$: VTAB 11: HTAB (39 - T2): PRINT CO$
 83  VTAB 1
 84  GOSUB 139
 85  VTAB 7: HTAB 1
 86  PRINT "WHICH IN OUR CASE IS:"
 87  VTAB 9: HTAB (39 - T): PRINT  SPC( T): VTAB 11: HTAB (39 - T2): PRINT
       SPC( T2)
 88  PRINT
 89  GOSUB 139
 90  VTAB 9: HTAB (39 - T): PRINT 8910: VTAB 11: HTAB (39 - T2): PRINT .39
      88
 91  VTAB 13: PRINT "BREAK EVEN POINT (YEAR)= 22342"
 92  GOSUB 139
 95  REM  END INSTRUCTION PORTION
 96  HOME : PRINT "NOW FOR YOUR BREAK-EVEN ANALYSIS"
 97  REM  BEGIN ANALYSIS PORTION
 98  PRINT : INPUT "PROJECTED SALES? ";PS
 99  INPUT "TOTAL FIXED COSTS? ";TF
 100  INPUT "TOTAL VARIABLE COSTS? ";TV
 101  LET A4 = 100 - (TV / PS * 100)
 102  LET RD = A4: GOSUB 137:A4 = RD
 103  FOR X = 1 TO 5: PRINT : NEXT X
 104  PRINT "CONTRIBUTION MARGIN=";A4;"%"
 105  PRINT "DO YOU WANT TO CHANGE THE CONTRIBUTION"
 106  PRINT "MARGIN %"
 107  VTAB 24: PRINT "ANSWER YES OR NO PLEASE";: GET A$
 108  IF A$ = "N" GOTO 110
 109  HOME : INPUT "THE CONTRIBUTION MARGIN % SHOULD BE: ";A4: GOTO 110
 110  HOME :BE = TF / (A4 * .01):RD = BE: GOSUB 137:BE = RD
 111  PRINT "BREAK EVEN POINT (YEAR)=";BE
 112  LET FC = TF / 254:RD = FC: GOSUB 137:FC = RD
 113  PRINT "FIXED COSTS PER DAY=$";FC
 114  PRINT "DAILY BREAK-EVEN POINT IS ";
 115  LET DB = ((TF / 254) / A4) * 100
 116  LET RD = DB: GOSUB 137:DB = RD
 117  PRINT "$";DB
 118  GOSUB 139
```

Listing continued.

swer to his questions are found in the Demonstration of Technique section of the computer program.

Computer Program and Input Data Preparation

An interactive computer program, written in Applesoft Basic and tested on an Apple II Plus, is provided in the program listing. A slight modification to the program may be necessary for users of other types of microcomputers. The program logic is based on the technical aspects of the break-even technique already discussed.

The program is completely self-documenting, so you can enter input data as required. A key feature is the provision for introducing the rudiments of break-even analysis, as demonstrated in the beauty shop "case example."

Once the program is entered and stored to disk, you then simply load the program into the computer and type RUN. The program then executes and

Listing continued.

```
119  HOME : PRINT "DO YOU WISH TO DO MORE (YES OR NO)";: GET A$: IF A$ =
     "Y" THEN 96
120  HOME
121  END : REM  END OF PROGRAM...SUBROUTINES FOLLOW
122  DATA  DEPRECIATION,1800
123  DATA  INSURANCE,210
124  DATA  RENT,4800
125  DATA  INTEREST,840
126  DATA  UTILITIES,900
127  DATA  LAUNDRY & MAINT.,360
128  DATA  TOTAL FIXED COSTS,8910
129  DATA  COSTS OF GOODS SOLD,2755
130  DATA  WAGES,20005
131  DATA  SUPPLIES,4800
132  DATA  PAYROLL TAXES,2015
133  DATA  ULILITIES,610
134  DATA  LAUNDRY & SHOP MAINTAINCE,610
135  DATA  TOTAL VARIABLE COST,30665
136  REM  ROUNDING ROUTINE
137  LET RD = (( INT (RD * 100 + .5)) / 100): RETURN
138  REM  PAUSE FOR KEYPRESS PAGE
139  VTAB 24: HTAB 5: PRINT "PRESS ANY KEY TO CONTINUE";: GET A$: RETURN
```

Fixed Costs		Variable Costs	
Depreciation	$1,800	Cost of Goods Sold	$ 2,775
Insurance	210	Wages	20,005
Rent	4,800	Supplies	4,800
Interest	840	Payroll Taxes	2,015
Utilities	900	Utilities	610
Laundry & Shop Maintenance	360	Laundry & Shop Maintenance	480
Total Fixed Costs	$8,910	Total Variable Costs	$30,665

Table 1. Sample of fixed and variable costs.

the title page is presented, followed by a brief introduction as to the use of break-even analysis. After pressing the space bar, the following menu appears:

1. DEMONSTRATION OF TECHNIQUE
2. BREAK-EVEN ANALYSIS CALCULATIONS

If you choose 1 and press the return key, the monitor displays both the fixed and variable costs from our "case example." Depressing the space bar again shows how the contribution margin is calculated, and hitting the space bar once more shows how the break-even point is calculated. Another space bar strike displays the break-even point formula in Figure 1.

The screen instructions ask you to hit any key; when you do, the formula appears with the appropriate "case example" figures. The answers appear at the bottom of the screen. At this point, any key press lets you calculate your break-even point with your own data input. The display screen prompts you with:

NOW FOR YOUR BREAK-EVEN ANALYSIS
PROJECTED SALES?
TOTAL FIXED COSTS?
TOTAL VARIABLE COSTS?

The menu asks for three pieces of information. Enter each as requested, and after entering the appropriate data, press the return key. The computer will then execute the formula and display the following answers:

CONTRIBUTION MARGIN = (Answer to your data)

At this point the program allows the user to change the contribution margin percentage if desired, or else type N for No. In the latter case, the program continues to calculate output. If the operator wants to change the calculated contribution margin percentage, the screen prints the following:

THE CONTRIBUTION MARGIN SHOULD BE: %

After striking the return key, the program calculates the appropriate break-even figures and displays them at the top of the screen.

For the output DAILY BREAK-EVEN POINT, we used 254 days of operation in the formulas at lines 112 and 115. However, if your business operates more days than this during the year, you'll have to change the figures in those lines to equal your number of operational days.

If you want to continue running further break-even analyses, the screen will prompt you at this point.

To bypass the Demonstration of Technique phase of the program, you can type in 2 from the original menu. The program will immediately prompt you for data input.

Additional Applications

This concept is also useful in determining the break-even point in terms of profit margin. Remember, the typed break-even approach develops the volume needed for producing no profit. What if you think you're in business to make a profit? (A common business malady!) Let's say you'd like a 10 percent profit margin on the project. Also, let's add that the original contribution margin for the project was 25 percent, but that's at zero profit. In effect, your 10 percent profit now acts like a variable cost. You must adjust contribution margin percentage accordingly: 25% − 10% = 15%. Now you can calculate the break-even point using the percentage of revenue approach:

$$BE = \frac{FC}{CM\% \text{ (new adjusted figure)}}$$

This gives you the profit margin built into your business. As in the computer program, just subtract the required profit margin (in percentage points) from the original contribution margin displayed on the monitor. Or, if you prefer, you can treat profit in dollar terms, and therefore as a fixed cost. If you decide to use this approach, you must add the dollar profit figure to the fixed cost category.

Another useful application of the break-even technique is in determining the sales dollars required to earn a given level of profit. Suppose a company wants a fixed dollar profit of $150,000. In this case, the profit is treated as a fixed cost. You must add it to the fixed cost established for the project, and calculate the break-even point as you've done before. The only difference would be the addition of the dollar figure to the fixed costs part of the data input.

A third application is finding the effect of an advertising budget. Advertising is essentially a fixed cost. Any added fixed costs raise the break-even

$$\text{BREAK-EVEN POINT} = \frac{\text{FIXED COSTS}}{\text{CONTRIBUTION MARGIN}}$$

Figure 1. The quintessential equation.

point of the business, and this requires added revenue (or lowered variable costs) to pay for them. The revenue for fixed costs comes from the contribution margin. For example, if the CM% is 25 percent, then four additional dollars of revenue are required to cover each additional dollar of fixed cost: $1 ÷ 25% = 4.

So, if the business is considering a $2,500 expenditure for an ad, it will need 4 × $2,500, or $10,000, in extra sales just to cover the cost of the advertisement. This approach provides the small business a built-in standard for judging the results of advertising. For example, if after an appropriate period added sales are not enough to justify the cost of the ad, the campaign can be abandoned.

A final interesting use of this technique lies in the area of expansion feasibility. Here, the analysis suggests whether or not to explore the expansion idea more carefully and in greater detail. In considering an expansion, one would have to estimate the appropriate increases in cost. With these costs taken into consideration, break-even analysis will tell you the amount of sales needed to break even.

> **"... break-even analysis will tell you the amount of sales needed to break even."**

Concluding Comments

Break-even analysis is useful for business planning. You can examine cost-volume-profit relationships under a variety of conditions. This aids in profit planning, advertising allocation, planning for new products, and expansion. You now have the technique that will help you attack new and novel business problems, and also help point you toward a rational decision. ■

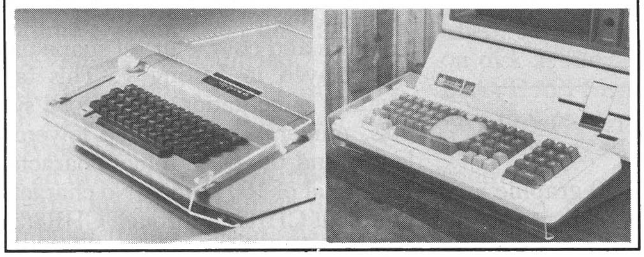
References

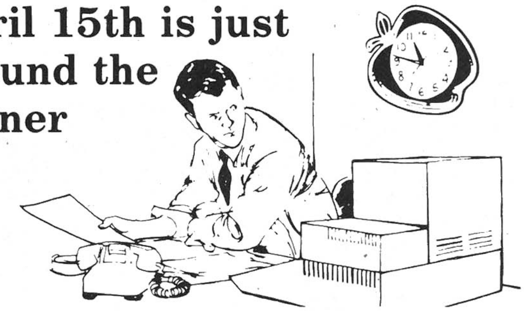
Droms, William G., *Finance and Accounting for Nonfinancial Managers*, Addison-Wesley Publishing Company, 1979.

Goulet, Peter G., "Attacking Business Decision Problems with Break-even Analysis," *Management Aids for Small Manufacturers*, MA#234, Small Business Administration, Washington, DC.

Hammel, Fred C., "Simple Break-even Analysis for Small Stores," *Small Marketers' Aids*, SMA#166, Small Business Administration, Washington, DC.

Steinhoff, Dan, *Small Business Management Fundamentals*, McGraw-Hill Book Company, 1978, pp. 199–209.





Apple/80

Your Apple will run those outstanding TRS-80 programs! A slight investment of your time, and the following tutorial, will enable you to convert thousands of TRS-80 programs to Applesoft. Baffle all your TRS-80 friends with your newfound expertise. . . and run their programs on your Apple.

by Hap Gaylord

How many times have I read a computer magazine and come upon a fantastic article, only to find that the material is applicable only for a TRS-80, and not my Apple? Frustrating? You bet. "Why not convert the program for my Apple?" I ask myself.

When I see the graphics I would need to rework to configure the program to my Apple, I sigh and turn to the next article.

Back up! The Apple has high-resolution graphics capability, doesn't it? Why not create a shape table that duplicates the TRS-80 character and graphics set? With only slight modification, the TRS-80 program would run on the Apple—graphics and all. Wow! A TRS-80 emulator—an Apple-80!

If you're familiar with TRS-80 Level II and Applesoft Basic, you know that conversion from one to the other requires changes to accommodate the subtle dialectic differences in the two systems. Fortunately, both Basics were written by Microsoft, so most logic, assignment statements, loops, and string manipulations are identical. Input/output routines will need modification, and several TRS-80 statements must be simulated for the Apple. Finally, hardware incompatibility creates a few real difficulties. The following article will demonstrate the conversion process step by step.

"With only slight modification, the TRS-80 program would run on the Apple—graphics and all. Wow! A TRS-80 emulator—an Apple-80!"

The Shape Table

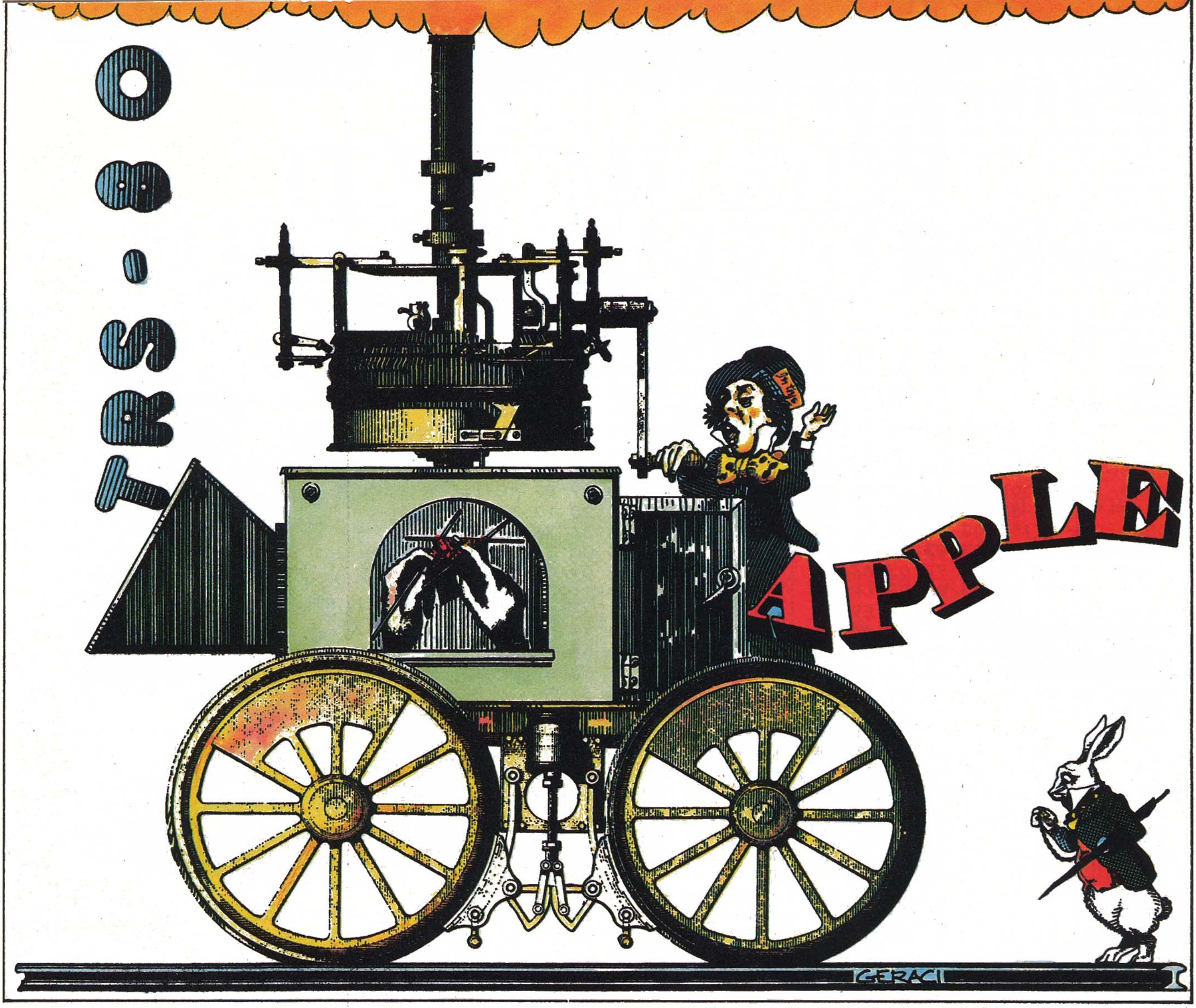
Recreating the TRS-80 graphic and character set in a shape table means using the Apple's 280 × 192 high-resolution screen to simulate the TRS-80's 64 by 16 line display. Dividing 280 by 64 reveals room for character spaces four dots wide. The extra 24-dot columns will be disregarded. The 16 lines on the TRS-80 match exactly the Apple's 192 vertical dots, if we use 12 dots per row. Thus, a full character space will be a 4 × 12 dot matrix.

One column of dots must separate characters, as well as rows of text. The ASCII set must be limited to a 3 × 8 dot matrix (see Figure 1). Such a narrow pattern seems impossible for alphabet design. However, only M, N, W, m, and w are troublesome. These five characters require double width. Any completely filled text line containing one or more of those letters will overflow, so the line has to be adjusted. That's a small price to pay to simulate the TRS-80 screen on the Apple. In the several programs I've converted, the overflow has occurred only once. The bonus of lowercase more than compensates for any inconvenience.

A hex dump of the complete TRS-80 graphics and character-set shape table is displayed in Listing 1. This table contains punctuation and special symbols, numerals, upper- and lowercase letters and the 64 graphics characters, for a total of 160 shapes. No characters represent CHR$(0) through CHR$(31), or Level II's space compression codes. Those values are ignored.

The shape table is fairly long, but you can enter it through the Apple's on-board monitor. If you prefer, I will supply a disk containing the binary files for $10 postpaid. To accomplish this from Basic, type CALL–151 and

Address correspondence to Hap Gaylord, PO Box 222, Bedford, NY 10506.



press return. A star prompt on your screen indicates your Apple is ready to accept input. Now type

800: C0 00 80 01 80 01 80 01 (RETURN)

for the first row of the shape table. Be sure to include the spaces as shown. Succeeding rows need no address, only the colon, as the Apple automatically stores the values successively. Type

: 80 01 80 01 80 01 80 01 (RETURN)

Continue entering each line until the shape table is completed. Double check your typing before pushing return. Errors are easy to commit with this type of data entry.

Wait to save the table to tape or disk until the machine-language data has been entered, in order to save all the binary segments as a single unit.

The three machine-language subroutines in Listings 2A, 2B and 2C print (actually draw) the TRS-80 alphanumeric characters on the high-resolution screen, handle the SET(X,Y) statement, and simulate POINT(X,Y). They could be drawn using Basic, but the process would be too slow because the Apple, in true high-resolution mode, creates graphics with 48 dots per character. The TRS-80 creates graphics with only 6 pixels (8 dots per pixel) per character space (see Figure 1).

Type the subroutines in from the hex dump (Listings 2) exactly the way you entered the shape table. When finished, save the entire binary segment on disk as file "APPLE-80" by typing BSAVE APPLE-80, A$800, L$B00. To save to tape, type 800.12DBW. From the monitor return to Basic with a control-B.

The shape table starts at 2048 ($800), the normal address for the beginning of an Applesoft program. Move any Basic program above the hires screen to 16385 with a

POKE 103,1:POKE 104,64:POKE 16384,0

before loading the program tape. The first two Pokes reset the program pointer. The last one places a zero in front of the program, a requirement of Applesoft.

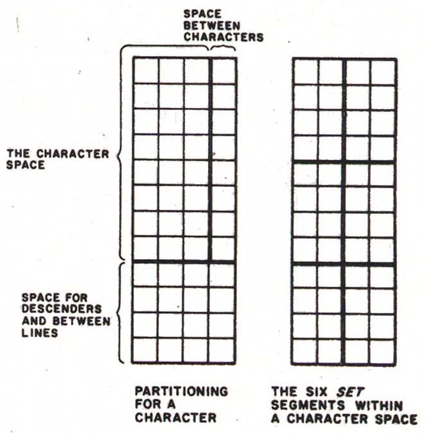


Figure 1. Character spacing.

Similarities

Table 1 lists the 37 statements

Program listing 1.
The shape table.

```
0800- 00 00 00 01 80 01 80 01
0808- 80 01 80 01 80 01 80 01
0810- 80 01 80 01 80 01 80 01
0818- 80 01 80 01 80 01 80 01
0820- 80 01 80 01 80 01 80 01
0828- 80 01 80 01 80 01 80 01
0830- 80 01 80 01 80 01 80 01
0838- 80 01 80 01 80 01 80 01
0840- 80 01 82 01 88 01 8D 01
0848- 9C 01 A5 01 AD 01 B3 01
0850- B8 01 BF 01 C5 01 CE 01
0858- D4 01 DB 01 DF 01 E5 01
0860- EC 01 F5 01 FD 01 06 02
0868- 0D 02 16 02 1F 02 28 02
0870- 2F 02 39 02 42 02 49 02
0878- 51 02 58 02 5E 02 65 02
0880- 6C 02 75 02 80 02 8A 02
0888- 92 02 9B 02 A4 02 AC 02
0890- B5 02 BF 02 C7 02 CF 02
0898- DA 02 E0 02 ED 02 FB 02
08A0- 04 03 0D 03 18 03 23 03
08A8- 2B 03 32 03 3C 03 45 03
08B0- 51 03 5B 03 64 03 6C 03
08B8- 73 03 7B 03 83 03 8B 03
08C0- 93 03 95 03 9D 03 A6 03
08C8- AD 03 B7 03 BF 03 C7 03
08D0- D3 03 DB 03 E1 03 E9 03
08D8- F2 03 F8 03 05 04 0D 04
08E0- 15 04 1F 04 29 04 2F 04
08E8- 37 04 3F 04 46 04 4D 04
08F0- 58 04 5F 04 6A 04 71 04
08F8- 79 04 7E 04 83 04 8A 04
0900- 90 04 92 04 97 04 9D 04
0908- A6 04 AD 04 B6 04 C1 04
0910- CE 04 D5 04 DF 04 E9 04
0918- F6 04 01 05 0E 05 1C 05
0920- 2D 05 35 05 3F 05 4B 05
0928- 59 05 64 05 71 05 7F 05
0930- 90 05 9B 05 A9 05 B7 05
0938- C8 05 D7 05 E8 05 FA 05
0940- 0F 06 18 06 23 06 2E 06
0948- 3D 06 48 06 56 06 64 06
0950- 76 06 81 06 8F 06 9D 06
0958- AE 06 BD 06 CE 06 E0 06
0960- F5 06 01 07 0F 07 1F 07
0968- 31 07 40 07 51 07 63 07
0970- 78 07 87 07 99 07 AB 07
0978- C0 07 D3 07 E8 07 FE 07
0980- 00 00 09 36 36 B6 04 00
0988- 36 0D 24 04 00 91 36 36
0990- 0F 18 0D 2D 17 24 24 15
0998- 3F 3F 04 00 31 FD 32 35
09A0- 35 1E AF 04 00 0D 36 1E
09A8- F6 36 0D 04 00 92 2D 17
09B0- 24 04 00 29 3E F6 04 00
09B8- 09 1E F6 76 76 04 00 0E
09C0- 76 F6 F6 04 00 12 2D 16
09C8- 3F 15 24 24 04 00 92 2D
09D0- 17 24 04 00 92 92 29 3E
09D8- F6 04 00 92 2D 04 00 92
09E0- 92 29 3E 04 00 09 F6 36
09E8- 1E 36 04 00 71 36 36 BE
09F0- 1C 24 24 24 00 2A 30 36
09F8- 36 36 6F 04 00 39 6E 36
0A00- 1E 17 36 2D 04 00 AD F6
0A08- 6F 32 F6 27 00 09 3E 1E
0A10- 2E 25 16 36 26 00 09 3F
0A18- 36 2E 0E 36 1E 27 00 09
0A20- BF 36 36 AE 0C E4 04 00
0A28- 2D 36 1E F6 36 04 00 F1
0A30- B6 36 15 0C 24 1C 0C 24
0A38- 00 F1 36 15 25 24 96 32
0A40- 26 00 11 35 B7 2A 3E 04
0A48- 00 91 35 B7 2A 3E F6 04
0A50- 00 52 F1 1E 15 15 04 00
0A58- 12 2D 16 3F 04 00 12 15
0A60- 15 17 17 04 00 62 15 F6
0A68- 36 16 04 00 91 2A 3E 37
0A70- 0D D7 0E 25 00 31 37 36
0A78- BD 36 0D 24 24 24 04 00
```

```
0A80- 75 BE 0E 36 17 27 24 24
0A88- 24 00 71 1F 36 36 AE 0C
0A90- 04 00 75 36 36 BE 27 24
0A98- 24 24 00 2D 9F 36 2D 9F
0AA0- 36 2E 25 00 2D 9F 36 2D
0AA8- 9F 36 26 00 71 1F 36 36
0AB0- AE 0C 24 27 00 36 2E 17
0AB8- 36 6E 24 24 24 24 00 2D
0AC0- 17 36 36 36 6F 04 00 2D
0AC8- 36 36 36 3E 1C 24 00 0D
0AD0- 36 17 76 36 1F 24 24 24
0AD8- 04 00 36 36 36 2E 25 00
0AE0- 36 36 36 6E 09 24 24 24
0AE8- BC 17 1C 04 00 15 AE AE
0AF0- AE 24 24 24 FC 9B 36 36
0AF8- 36 04 00 71 36 36 BE 1C
0B00- 24 24 24 00 75 36 17 97
0B08- 22 24 24 24 00 F1 36 36
0B10- AE 0D 1C 3C 0C 24 24 00
0B18- 75 36 17 0E 36 1F 24 24
0B20- 24 04 00 29 D7 AE 15 36
0B28- 17 27 00 2D 17 36 36 36
0B30- 04 00 36 36 36 2E 25 24
0B38- 24 24 04 00 36 36 95 22
0B40- 64 24 24 04 00 36 36 36
0B48- 66 0C B4 71 24 24 24 24
0B50- 00 36 96 36 0D 24 1C 64
0B58- 24 04 00 36 B6 52 24 24
0B60- 25 24 04 00 2D 36 17 BE
0B68- 36 2D 04 00 A9 3F 15 36
0B70- 36 26 00 31 36 36 3E 0D
0B78- 17 04 00 52 F1 1E 2D 17
0B80- 0E 04 00 12 0E 0E 3F 15
0B88- 17 04 00 92 92 92 2D 25
0B90- 00 00 00 00 00 92 75 3E
0B98- 1E 2E 25 04 00 36 2E 0E
0BA0- 36 17 27 24 04 00 92 09
0BA8- F7 36 15 25 00 09 36 3E
0BB0- 1E 36 15 25 24 04 00 92
0BB8- 71 1F 2E B5 E7 04 00 71
0BC0- 1F 36 2E 17 36 04 00 92
0BC8- F1 36 15 25 24 96 32 1F
0BD0- 15 04 00 36 2E 17 36 6E
0BD8- 24 24 00 11 16 36 36 04
0BE0- 00 89 16 36 36 F6 1E 04
0BE8- 00 36 36 0D 17 0E FE 24
0BF0- 04 00 31 36 36 36 04 00
0BF8- 92 0D 0D 75 36 FE 23 24
0C00- 1F 37 36 04 00 92 2E 35
0C08- 36 1F 24 04 00 92 71 36
0C10- 17 1C 24 04 00 92 75 36
0C18- 17 97 22 24 24 24 00 92
0C20- 32 AE 95 22 24 24 24 27
0C28- 00 92 2E FD 32 26 00 92
0C30- 29 D7 15 0E 17 27 00 31
0C38- 3E 0D 17 36 36 04 00 92
0C40- 36 AE 25 24 24 00 92 36
0C48- 95 64 24 04 00 92 36 AE
0C50- 65 24 56 2A 0C 24 24 00
0C58- 92 B6 6E E4 0C 24 00 92
0C60- 36 AE 25 24 B4 92 BE 1C
0C68- 04 00 92 2D BE 1E 2E 25
0C70- 00 09 1E 36 AF 36 15 04
0C78- 00 31 1E 6E 24 00 31 B6
0C80- 32 26 00 0E 36 BD 36 1E
0C88- 04 00 36 0E 76 36 04 00
0C90- 00 00 36 2E 24 24 00 09
0C98- 36 2E 24 24 00 2D 35 3F
0CA0- 37 2D 35 3F 27 00 92 32
0CA8- 36 25 24 04 00 36 36 36
0CB0- 2E 24 24 24 24 00 92 32
0CB8- 36 25 24 0C 24 2C 36 26
0CC0- 00 36 36 36 2E 24 24 24
0CC8- 2C 36 2E 24 24 00 92 4A
0CD0- 36 2E 24 24 00 35 37 35
0CD8- 6F 32 36 25 24 04 00 09
0CE0- 36 36 36 2E 24 24 24 24
0CE8- 00 36 2E 24 2C 36 36 36
```

Listing 1 continued on page 101.

TRS-80 Level II and Applesoft have in common. Any time these statements appear in a program being converted, type them into the Apple exactly as written for the TRS-80. They'll make up more than half of any TRS-80 program, so much of a converted listing will need absolutely no changes.

Output

An important difference between the two Basics occurs in printing to the video screen. The print statement is almost never used in converted Apple-80 programs, since all text and graphics are drawn in high resolution. This is easier than it sounds. All the machine-language print subroutine requires is to assign the starting screen location (H8%), define the string to be printed (M$), and call the hires draw routine (CALL 4200).

The TRS-80 print statements come in several varieties. Table 2 shows the major types together with their translations. For example

PRINT@ xxxx,M$

where xxxx is a number between 0 and 1023, means "PRINT M$ beginning at the screen location number xxxx."

For those unfamiliar with the TRS-80, the screen location numbers begin with 0 in the upper left corner, and proceed left to right, 64 characters across, line by line, down the screen, for 16 lines. Thus, the top line runs from 0 to 63, the second line from 64 to 127, and so on. For our Apple conversion, we use the line

H8% = xxxx:M$ = "(the string to be printed)": CALL 4200

The xxxx is the same number that followed the @ in the TRS-80 program. It is the beginning print position on the screen. If the TRS-80 program gives no screen number, compute and supply the correct video location for the Apple conversion. Drawing and numbering a sketch of the 64 by 16 screen will aid placement.

The print routine normally sends lowercase letters to the screen, but any letter preceded by a Λ (typed with a shift-N) will print uppercase. You must designate each capital letter separately.

Any numerical variable or number to be printed must first be converted to a string with the STR$ function. (See



. . . is what you can always expect from TG Products' game controls. They combine quality components and exclusive features to give you the precision control you need to add to the fun and excitement of game play on your personal computer. (Compatible with Atari, Apple and IBM.)*

TG Products, the oldest and largest manufacturer of accessories for personal computers, brings products to you that are designed to provide the ultimate in man-machine interface, and years of gaming fun.

SELECT-A-PORT extends the flexibility of the Apple's game port. It has three switch selectable sockets isolated by diodes to prevent interference, a socket which automatically modifies the Joystick, Paddles or Track Ball to operate as the second unit in dual unit games, and one socket with no isolation for highly sensitive devices.

JOYSTICK has completely linear operation, exclusive trim adjustments for both axes and a self-centering feature you will want to use for some games and disengage for others.

GAME PADDLES have large, easy to push buttons and extra long 60″ cords which make them easier to handle and add to your game playing enjoyment.

TRACK BALL is a new omni-directional guidance system which adds excitement and quick response to games requiring a lot of movement commands. It has a lower control-to-movement ratio than the Joystick control which allows more sensitive positioning for graphics work on the screen. Fire control buttons are recessed below the ball control plane for unobstructed movement.

Ask for these fine control products at your dealer or order direct from TG Products, 1104 Summit Ave., Suite 110, Plano, Texas 75074 or call (214) 424-8568.

TG
PRODUCTS

*Atari, Apple and IBM are registered trademarks of Atari Inc., Apple Computer Inc., and International Business Machines Corporation, respectively.

All mail orders add $2.00 for postage and insurance.
Texas residents add 5% sales tax. Allow two weeks for shipment.

Circle 47 on Reader Service card.

Table 2 for an example.) Any semicolon or comma in a converted print statement must be omitted, but figured into the H8% calculation. The semicolon can be approximated by H8% = H8% + LEN(M$). However, watch out for double-width letters and the capital letter Λ marker. The comma can be simulated by successively assigning to H8% the value of the first space in each column.

The machine-language routine will move down a row automatically after filling a line, but forgets where on the screen it finished printing the previous string. The screen will overwrite rather than scroll. Scroll is a function rarely used now, except in some games. The hires draw routine will move automatically to the top of the screen after using the final lower right corner space.

Often characters are poked onto the TRS-80 video screen instead of being printed. Knowing that the screen begins at memory location 15360 and runs line by line consecutively down to 16383, you need only subtract the 15360 to obtain the PRINT@ numbers. Then our Apple-80 print routine will work easily with M$ = CHR$(N), where N is the number to be poked.

Examine the sample in Table 2. If the Poke address is out of the range 15360 to 16383, it's not a Poke to the screen and must be handled differently, as we'll learn later.

Listing 1 continued from page 98.

```
0CF0- 2E 24 24 24 24 00 92 2A
0CF8- 2D 3E 3F 2E 2D 3E 3F 04
0D00- 00 35 37 35 37 2D 35 3F
0D08- 37 2D 35 3F 27 00 09 35
0D10- 37 35 AF 3F 37 2D 35 3F
0D18- 37 2D 25 00 2D 35 3F 37
0D20- 2D 35 3F 37 2D 35 3F 37
0D28- 2D 35 3F 27 00 92 92 12
0D30- 36 2E 24 24 00 35 37 35
0D38- B7 92 36 2E 24 24 00 09
0D40- 35 37 35 B7 92 33 36 27
0D48- 24 04 00 2D 35 3F 37 2D
0D50- 35 3F B7 92 36 2E 24 24
0D58- 00 92 32 36 36 36 25 24
0D60- 24 24 04 00 36 36 36 36
0D68- 36 2E 24 24 24 24 24 24
0D70- 00 09 35 37 35 F7 36 36
0D78- 36 3E 24 24 24 24 00 2D
0D80- 35 3F 37 2D 35 3F 37 36
0D88- 36 36 2E 24 24 24 24 00
0D90- 89 92 35 37 35 F7 37 35
0D98- 37 25 00 35 37 35 6F 2A
0DA0- 3E 2E 3E 17 37 35 37 25
0DA8- 00 09 35 37 35 37 35 37
0DB0- 35 BF 37 35 37 25 00 36
0DB8- 2E 24 2C 35 37 35 37 35
0DC0- 37 35 BF 36 3E 24 24 00
0DC8- 92 2A 2D 3E 3F 2E 2D 3E
0DD0- 3F 36 36 25 24 04 00 35
0DD8- 37 35 37 2D 35 3F 37 2D
0DE0- 35 3F 37 36 2E 24 24 00
0DE8- 09 35 37 35 37 36 2E 24
0DF0- FC 36 36 36 3E 24 24 24
0DF8- 24 00 2D 35 3F 37 2D 35
0E00- 3F 37 2D 35 3F 37 2D 35
0E08- 3F 37 36 2E 24 24 00 09
0E10- 92 92 12 36 2E 24 24 00
0E18- 35 37 35 6F 92 12 36 2E
0E20- 24 24 00 09 35 37 35 B7
0E28- 92 36 2E 24 24 00 2D 35
0E30- 3F 37 2D 35 3F 6F 92 12
0E38- 36 2E 24 24 00 92 2A 3E
0E40- 2E 3E 8D 35 37 35 27 00
0E48- 35 37 35 37 35 37 35 6F
0E50- 2A 3E 2E 3E 04 00 09 35
0E58- 37 35 F7 37 35 37 AD 35
0E60- 37 35 27 00 2D 35 3F 37
0E68- 2D 35 3F 37 35 37 35 6F
0E70- 2A 3E 2E 3E 04 00 89 92
0E78- 36 36 36 2E 24 24 24 24
0E80- 00 35 37 35 6F 32 36 36
0E88- 36 25 24 24 24 04 00 09
0E90- 36 36 36 36 36 2E 24 24
0E98- 24 24 24 24 00 36 2E 24
0EA0- 2C 36 36 36 36 36 2E 24
0EA8- 24 24 24 24 24 00 92 32
0EB0- 36 25 24 35 36 36 36 25
0EB8- 24 24 24 04 00 35 37 35
0EC0- 37 36 2E 24 2C 36 36 36
0EC8- 2E 24 24 24 24 00 09 35
0ED0- 37 35 AF 3F 37 2D 35 3F
0ED8- 37 2D 35 36 3E 24 24 00
0EE0- 36 36 36 2E 24 24 24 2C
0EE8- 36 36 36 36 36 2E 24 24
0EF0- 24 24 24 24 00 92 92 12
0EF8- 2D 35 3F 37 2D 35 3F 27
0F00- 00 35 37 35 B7 92 2D 35
0F08- 3F 37 2D 35 3F 27 00 09
0F10- 35 37 35 AF 92 3A 3F 2E
0F18- 2D 3E 3F 2E 2D 04 00 2D
0F20- 35 3F 37 2D 35 3F B7 92
0F28- 2D 35 3F 37 2D 35 3F 27
0F30- 00 92 2A 3E 2E 3E 2E 2D
0F38- 3E 3F 2E 2D 3E 3F 04 00
0F40- 35 37 35 37 35 37 35 37
0F48- 2D 35 3F 37 2D 35 3F 27
0F50- 00 09 35 37 35 BF 37 35
0F58- 37 BD 2D 35 3F 37 2D 35
0F60- 3F 27 00 2D 35 3F 37 2D
0F68- 35 3F 37 35 37 35 37 2D
0F70- 35 3F 37 2D 35 3F 27 00
0F78- 89 92 35 37 35 AF 3F 37
0F80- 2D 35 3F 37 2D 25 00 35
0F88- 37 35 6F 2A 3E 2E 3E 15
0F90- 3F 37 2D 35 3F 37 2D 25
0F98- 00 09 35 37 35 37 35 37
0FA0- 35 AF 3F 37 2D 35 3F 37
0FA8- 2D 25 00 36 2E 24 2C 35
0FB0- 37 35 37 35 37 35 AF 3F
0FB8- 37 2D 35 3F 37 2D 25 00
0FC0- 92 2A 2D 3E 3F 2E 2D 3E
0FC8- 3F 2E 2D 3E 3F 2E 2D 3E
0FD0- 3F 04 00 35 37 35 37 2D
0FD8- 35 3F 37 2D 35 3F 37 2D
0FE0- 35 3F 37 2D 35 3F 27 00
0FE8- 09 35 37 35 AF 3F 37 2D
0FF0- 35 3F 37 2D 35 3F 37 2D
0FF8- 35 3F 37 2D 25 00 2D 35
1000- 3F 37 2D 35 3F 37 2D 35
1008- 3F 37 2D 35 3F 37 2D 35
1010- 3F 37 2D 35 3F 27 00 00
```

Input

The Input statement won't print an imbedded quote using Apple-80. The imbedded string must be printed separately. In fact, avoid Input entirely since any typed characters would not be echoed to the Apple's high-resolution screen for printing. Input from the keyboard is handled best with Get or Peek(– 16384). Use Get to replace Input in a routine similar to the one in Program listing 4, lines 9 to 29. A Gosub 9 invokes the Get command and, on return, B$ holds the input string.

This string can be reassigned to the TRS-80 variable name. With Get, we can fetch a character, then print the character on the high-resolution screen, and accommodate backspaces. Each typed character is concatenated into the final string.

To type capital letters, first type Λ, then the letter. The Λ won't show on your screen, but will be placed into the string. If a number is typed, the VAL function will evaluate the resulting string in the body of the program.

Get causes the computer to wait until a character is typed. Use PEEK (– 16384) for input on the fly. The value returned by the Peek will be larger than 127 if a key was pressed. This is the Apple equivalent of the TRS-80's INKEY$. Table 3 compares the various input routines.

TRS-80 games often use another form of input. PEEK(N), where 14336 <N<15359, is a running look at the keyboard switch network to see whether any key is currently depressed. For the Apple, use N = PEEK (– 16384) for this particular function. Avoid resetting the keyboard strobe with POKE – 16368,0. Each key pressed subsequently will change the value of N, but N will remain fixed until you press another key.

Unfortunately, the two machines differ here. On the TRS-80, releasing all keys resets N to zero. On the Apple, however, the old value of N remains until reset with POKE – 16368,0. This is true even if no keys remain depressed. On the other hand, if the program executes a POKE – 16368,0 and you continue to depress a key, the Apple treats the key as released unless the repeat key is also depressed, or the key is released and depressed again.

Program translations have to be designed around this difference. For example, one could use I, J, K, L and M for up, left, stop, right and down, re-

Program listing 2.
Hex dump.

```
1067- 60
1068- A9 FE 85 EB A9 0F 85 EC
1070- A9 00 85 FE A0 04 B1 69
1078- 85 07 88 B1 69 85 06 88
1080- B1 69 F0 E3 85 FF A0 09
1088- B1 69 85 09 C8 B1 69 85
1090- 08 29 C0 85 F9 A5 09 4A
1098- 66 F9 4A 66 F9 66 F9 A5
10A0- F9 4A 65 F9 85 09 A5 08
10A8- 29 3F 0A 0A 85 08 A4 FE
10B0- A9 20 85 FE A9 00 85 EF
10B8- A5 E8 85 FC A5 E9 85 FD
10C0- B1 06 C9 20 90 0C C9 C0
10C8- B0 08 C9 5E D0 0B A9 00
10D0- 85 FE C8 C5 FF F0 90 D0
10D8- E7 C9 41 90 07 C9 5B B0
10E0- 03 18 65 FE 84 FE 85 FB
10E8- 0A 90 03 E6 FD 18 65 FC
10F0- 90 02 E6 FD 85 FC A0 00
10F8- B1 FC 85 F9 C8 B1 FC 85
1100- FA A5 E8 18 65 F9 90 02
1108- E6 FA 85 FC A5 FA 18 65
1110- E9 85 FD A5 FB C9 4D F0
1118- 12 C9 4E F0 0E C9 57 F0
1120- 0A C9 6D F0 06 C9 77 F0
1128- 02 D0 04 A9 01 85 EF A2
1130- 00 20 F0 F6 A5 09 A6 08
1138- A0 00 20 11 F4 A6 EB A4
1140- EC A9 00 20 01 F6 A5 EF
1148- F0 25 A5 08 85 ED A5 09
1150- 85 EE 20 A0 11 A5 09 A6
1158- 08 A0 00 20 11 F4 A6 EB
1160- A4 EC A9 00 20 01 F6 A5
1168- ED 85 08 A5 EE 85 09 A2
1170- 03 20 F0 F6 A5 09 A6 08
1178- A0 00 20 11 F4 A6 FC A4
1180- FD A9 00 20 01 F6 20 A0
1188- 11 A5 EF F0 03 20 A0 11
1190- A4 FE C8 C4 FF F0 03 4C
1198- B0 10 60 00 00 00 00 00
11A0- A5 08 18 69 04 85 08 D0
11A8- 0D A5 09 18 69 0C C9 C0
11B0- 90 02 A9 00 85 09 60
*
```

Listing 2A.

```
]CALL-151

*1200.1235

1200- 88 B1 69 0A 0A 85 1F A0
1208- 00 20 57 F4 A4 1F C8 C8
1210- C8 A5 1E A2 00 20 3A F5
1218- E6 1E A6 1E A5 1F A0 00
1220- 20 57 F4 A4 1F C8 C8 C8
1228- A5 1E A2 00 20 3A F5 60
```

Listing 2B.

```
125C- A9 04 85 EF
1260- A9 00 85 1F A0 0A B1 69
1268- 85 1E 0A AA 88 B1 69 0A
1270- 0A A0 00 20 11 F4 A5 1E
1278- C9 07 90 0B EE 26 00 EE
1280- 26 00 38 E9 07 B0 F1 A8
1288- F0 14 C9 04 90 0B EE 26
1290- 00 38 E9 03 0A E9 01 B0
1298- 05 C9 03 F0 23 0A 85 1E
12A0- A6 1E A0 00 B1 26 4A CA
12A8- 30 02 10 FA 26 1F 4A 26
12B0- 1F A5 27 18 69 04 85 27
12B8- EA EA C6 EF F0 1D D0 E0
12C0- A0 00 B1 26 0A 0A 26 1F
12C8- C8 B1 26 4A 26 1F A5 27
12D0- 18 69 04 85 27 EA EA C6
12D8- EF D0 E5 60
```

Listing 2C.

spectively. When K, the stop key, is pressed, do a POKE – 16368,0 to reset the keyboard.

Finally, the TRS-80 has different Peek addresses for different sets of keys, and varying values for each separate key. The Apple uses ASCII numbers added to 128. See Table 6 for comparable values, and the example in Table 5.

Other Differences

Differences between Level II Basic and Applesoft can be reconciled by correcting the syntax to satisfy Applesoft, or by subroutines that accomplish the same purpose. CLS (Clear the Screen), for instance, can be simulated by HGR:POKE – 16302,0:H8% = 0. Since CLS occurs so often in programs, I constructed it as a Basic subroutine, as in Program listing 5. A simple Gosub 5 will clear the screen and prepare to print at the top left corner.

Perhaps the most important difference in the two Basics is the absence of the Else statement in Applesoft. Else gives the TRS-80 the capability of including both true and false actions in If-Then comparisons. The statements

```
100 CLEAR 200:CLS:I=1:DIM N$(12),H(12):GOSUB 3000
110 PRINT @ 0,"This bar graph can represent up to 12 quantities,"
120 PRINT"with values from 1 to 10 for each."
130 PRINT"You can either use the default quantities supplied with this"
140 PRINT"program, or type in your own.  Will you use mine (1) or"
150 PRINT"yours (2)? ";
160 K$=INKEY$:IF K$="" THEN 160
170 IF K$="1" THEN GOSUB 1000:GOTO 500
180 IF K$<>"2" THEN 160
190 CLS:GOSUB 3000
200 PRINT @ 0,"What will the title of your bar graph be";
210 INPUT T$
220 IF LEN(T$)=0 THEN T$="Bar Graph"
230 CLS:GOSUB 3000
240 PRINT @ 0,"Name";I;"(or enter 0 to end) ";
250 INPUT N$(I):IF N$(I)="0" THEN N$(I)="":GOTO 500
260 IF LEN(N$(I))=0 THEN 240
270 PRINT @ 0,"What is the value for ";N$(I);" (1-10) ";
280 INPUT H(I):IF H(I)<0 OR H(I)>10 THEN 270
290 IF LEN(N$(I))>4 THEN N$(I)=LEFT$(N$(I),4)
300 PRINT @ 0,STRING$(63,32)
310 IF I=12 THEN 500 ELSE I=I+1:GOTO 240
500 CLS:GOSUB 3000
510 PRINT @ 96-LEN(T$)/2,T$
520 FOR A=1 TO 12:W=10*A-4
530 PRINT @ 963+5*(A-1),N$(A);
540 IF H(A)=0 THEN 590
550 FOR Y=1 TO H(A)*3
560 FOR X=W TO W+5
570 SET(X,45-Y)
580 NEXT X,Y
590 NEXT A
600 GOTO 600
1000 FOR I=1 TO 12:READ N$(I),H(I):NEXT I
1010 T$="Average Number of Pupils Absent Each Day"
1020 RETURN
1030 DATA Jan,10,Feb,11,Mar,8,Apr,5,May,2,Jun,2,Jul,0,Aug,0,Sep,1,Oct,2,Nov,5,Dec,9
3000 FOR X=4 TO 123:SET(X,44):NEXT X
3010 FOR Y=10 TO 1 STEP-1:PRINT@ 64*Y+256,11-Y;:NEXT Y
3020 RETURN
```

Program listing 3. TRS-80 bar graph.

ABS	EXP	MID$	RETURN
AND	FOR-TO-STEP	NEXT	RIGHT$
ASC	GOSUB	NOT	SGN
ATN	GOTO	ON GOSUB	SIN
CHR$(except 1–31, 192–255)	IF-THEN	ON GOTO	SQR
	INT	OR	STOP
COS	LEFT$	READ	STR$
DATA	LEN	REM (' also means REM)	TAN
DIM	LET		VAL
END	LOG	RESTORE	

Table 1. Applesoft and TRS-80 common statements.

```
0 M$ = "":H8% = 0
1 D$ =  CHR$ (13) +  CHR$ (4): IF
     PEEK (104) <  > 64 THEN  POKE
    103,1: POKE 104,64: POKE 163
    84,0: PRINT D$;"RUN APPLE-80
     BAR GRAPH"
2  HCOLOR= 3: ROT= 0: SCALE= 1: POKE
    232,0: POKE 233,8: PRINT D$;
    "BLOAD APPLE-80 TABLE+PRINT"

4  HOME : GOTO 100
5  HGR : POKE  - 16302,0:H8% = 0:
     RETURN
9 B$ = ""
10 M$ =  CHR$ (95): CALL 4200: GET
    A$: IF A$ <  >  CHR$ (8) THEN
    20
12  IF  LEN (B$) = 1 THEN B$ = ""
    : GOTO 16
14  IF  LEN (B$) = 0 THEN 10
15 A$ =  RIGHT$ (B$,1):B$ =  LEFT$
     (B$, LEN (B$) - 1)
16 M$ = " ": IF A$ = "M" OR A$ =
    "W" OR (A$ = "N" AND  RIGHT$
    (B$,1) = "^") THEN M$ = "  "
    :H8% = H8% - 1
17  CALL 4200:H8% = H8% - 1: IF B
    $ = "^" THEN 9
18  IF  RIGHT$ (B$,1) = "^" THEN
    B$ =  LEFT$ (B$, LEN (B$) -
    1)
19  GOTO 10
20  IF A$ =  CHR$ (13) THEN M$ =
    " ": CALL 4200: RETURN
21  IF  ASC (A$) < 32 THEN 10
22  IF A$ = "^" AND  RIGHT$ (B$,1
    ) <  > "^" THEN B$ = B$ + A$
    : GOTO 10
23  IF A$ = "^" THEN 10
24  IF  ASC (A$) > 64 AND  ASC (A
    $) < 91 AND  RIGHT$ (B$,1) =
    "^" THEN M$ = "^" + A$: GOTO
    28
25  IF  RIGHT$ (B$,1) = "^" AND (
      ASC (A$) < 65 OR  ASC (A$) >
    90) THEN 10
26 M$ = A$
28  CALL 4200:H8% = H8% + 1: IF  LEN
    (B$) = 0 THEN 30
29  IF A$ = "M" OR A$ = "W" OR (A
    $ = "N" AND  MID$ (B$, LEN (
    B$),1) = "^") THEN H8% = H8%
     + 1
30 B$ = B$ + A$: GOTO 10
100  GOSUB 5:I = 1: DIM N$(12),H(
    12): GOSUB 3000
110 H8% = 0:M$ = "^THIS BAR GRAPH
     CAN REPRESENT UP TO 12 QUAN
    TITIES,": CALL 4200
120 H8% = 64:M$ = "WITH VALUES FR
    OM 1 TO 10 FOR EACH.": CALL
    4200
130 H8% = 128:M$ = "^YOU CAN EITH
    ER USE THE DEFAULT QUANTITIE
    S SUPPLIED WITH THIS": CALL
    4200
140 H8% = 192:M$ = "PROGRAM, OR T
    YPE IN YOUR OWN.  ^WILL YOU
    USE MINE (1) OR": CALL 4200
150 H8% = 256:M$ = "YOURS (2)? ":
     CALL 4200
160  GET K$
170  IF K$ = "1" THEN  GOSUB 1000
    : GOTO 500
180  IF K$ <  > "2" THEN 160
190  GOSUB 5: GOSUB 3000
200 H8% = 0:M$ = "^WHAT WILL THE
    TITLE OF YOUR BAR GRAPH BE?
    ": CALL 4200
210 H8% = 44: GOSUB 9:T$ = B$
220  IF  LEN (T$) = 0 THEN T$ = "
    ^BAR ^GRAPH"
230  GOSUB 5: GOSUB 3000
240 H8% = 0:M$ = "^NAME " +  STR$
    (I) + " (OR ENTER 0 TO END)
    ": CALL 4200
250 H8% = 31: GOSUB 9:N$(I) = B$:
     IF N$(I) = "0" THEN N$(I) =
    "": GOTO 500
260  IF  LEN (N$(I)) = 0 THEN 240
270 H8% = 0:M$ = "^WHAT IS THE VA
    LUE FOR " + N$(I) + " (1-10)
      ": CALL 4200
280 H8% = 31 +  LEN (N$(I)): GOSUB
    9:H(I) =  VAL (B$): IF H(I) <
    0 OR H(I) > 10 THEN 270
290  IF  LEN (N$(I)) > 4 THEN N$(
    I) =  LEFT$ (N$(I),4)
295  IF  RIGHT$ (N$(I),1) = "^" THEN
    N$(I) = N$(I) + " "
300 M$ = "": FOR J = 1 TO 63:M$ =
    M$ +  CHR$ (32): NEXT J:H8% =
    0: CALL 4200
310  IF I = 12 THEN 500
320 I = I + 1: GOTO 240
500  GOSUB 5: GOSUB 3000
510 H8% = 96 -  LEN (T$) / 2:M$ =
    T$: CALL 4200
520  FOR A = 1 TO 12:W = 10 * A -
    4
530 H8% = 963 + 5 * (A - 1):M$ =
    N$(A): CALL 4200
540  IF H(A) = 0 THEN 590
550  FOR Y = 1 TO H(A) * 3
560  FOR X = W TO W + 5
570 H8% = X + 256 * (45 - Y): CALL
    4600
580  NEXT X,Y
590  NEXT A
600  END
1000  FOR I = 1 TO 12: READ N$(I)
    ,H(I): NEXT I
1010 T$ = "^AVERAGE ^NUMBER OF ^P
    UPILS ^ABSENT ^EACH ^DAY"
1020  RETURN
1030  DATA ^JAN,10,^FEB,11,^MAR,8
    ,^APR,5,^MAY,2,^JUN,2,^JUL,0
    ,^AUG,0,^SEP,1,^OCT,2,^NOV,5
    ,^DEC,9
3000  FOR X = 4 TO 123:H8% = X +
    256 * 44: CALL 4600: NEXT X
3010  FOR Y = 10 TO 1 STEP  - 1:H
    8% = 64 * Y + 256:M$ =  STR$
    (11 - Y): CALL 4200: NEXT Y
3020  RETURN
```

Program listing 4. Apple-80 conversion.

immediately following the Then up to the Else are executed if true; the statements following the Else are performed if false.

Every If-Then-Else must be split into two lines on the Apple. The first contains the If-Then up to, but excluding, the Else. The second contains the statements following the Else. Additionally, the new If-Then must include a jump over the newly created second line to avoid doing the Else statements when the computer automatically moves on to the next program line. For example,

```
100 IF C>350 THEN T = T + 1 ELSE C = C + 1
120 (program continues)
```

would be rewritten as

```
100 IF C>350 THEN T = T + 1:GOTO 120
110 C = C + 1
120 (program continues)
```

Also examine lines 310 and 320 of Listing 4 for another variation.

More Differences

The TRS-80's random number generator, RND(N), must be converted to Applesoft syntax INT(RND(1)*N) + 1, where N is a positive integer. (Note, however, that RND(0) on the TRS-80 is identical to RND(1) on the Apple. Both produce decimals between 0 and 1.)

STRING$ doesn't exist in Apple's Basic. On the TRS-80, it creates a multiple line of the character whose ASCII value is designated in the parenthetical argument. For example, STRING$ (20,65) would be a string of 20 A's (65 is the ASCII value for A). We can simulate this function by simply typing the 20 A's between quotation marks, or with a loop to build up the string, as in M$ = "":FOR I = 1 TO 20:M$ = M$ + CHR$(65):NEXT I.

The latter approach would be required if the 20 were a variable, or if the character were a TRS-80 graphics character (CHR$(128) through CHR$(191)).

SET(X,Y) and RESET(X,Y), which create the TRS-80's graphics (similar to Apple's Plot command), are simulated by a Call to a machine-language routine (CALL 4600). The X (0 to 127 across) and Y (0 to 47 down) coordinates are passed to the routine through H8%, so a converted line looks like H8% = X + 256*INT(Y):CALL 4600. Unless you know Y is an integer you must use INT(Y), because multiplying Y by 256 may magnify some of the decimal enough to corrupt the integer. If the HCOLOR has been set to white, the above line will Set; if the HCOLOR is black, it will Reset (i.e., erase). (See line 570 of Listing 4.) One slight difference here is that this routine doesn't erase a character when Set overlays it. The TRS-80 does. This strikes me as an advantage for overprinting.

POINT(X,Y) requires a special machine-language subroutine as well. This function determines whether a specified point (X across, Y down) is currently set or not, similar to Apple's SCRN(X,Y). A CALL 4700 invokes the subroutine, and the result can be tested with a PEEK(31). Only if PEEK(31) equals 255 is the point set. The entire Applesoft line to match IF POINT (X,Y) THEN. . . or IF POINT(X,Y) = – 1 THEN. . . would be

```
H8% = X + 256*INT(Y):CALL 4700:
IF PEEK(31) = 255 THEN. . .
```

DEFINT (DEFine INTeger) and DEFSTR (DEFine STRing) are not available as Applesoft statements. Instead, the Apple uses the % (integer) or

Table 2.
Print statements.

TRS-80	APPLE-80
PRINT@309,"SCORE: "	H8% = 309:M$ = "^S^C^O^R^E: ": CALL 4200
PRINT@0,CHR$(191);CHR$(65)	H8% = 0:M$ = CHR$(191) + CHR$(65): CALL 4200
PRINT "Rules"	H8% = xxxx:M$ = "^RULES":CALL 4200 (where you must supply the number for xxxx)
PRINT@K, 472	H8% = K:M$ = "472":CALL 4200
PRINT@K, N	H8% = K:M$ = STR$(N):CALL 4200
100 PRINT "You Win! ";:GOTO 100	100 M$ = "^YOU ^WIN! ":H8% = H8% + LEN(M$) – 1:CALL 4200:GOTO 100
POKE 15360 + N,42	H8% = N:M$ = CHR$(42):CALL 4200

$ (string) on every occurrence of the variable's name. For example, 10 DEFINT A : DEFSTR T on the TRS-80 would require an Apple program to use A% and T$ every time those variable names were used. Line 10 above can't be duplicated in an Apple program. When converting a TRS-80 program, remember that DEFSTR A means any variable name anywhere in the program that begins with an A is a string, whether a dollar sign follows the name or not. Thus, A2 = "I WIN!" is perfectly acceptable on the TRS-80. DEFSTR A–T assigns all variables beginning with the letters from A through T to be strings. Any later individual assignment using the symbols % (integer), ! (single precision), or $ (string) on each occurrence of the name will override the DEFSTR A–T temporarily.

DEFSNG (DEFine SiNGle precision) matches the default for the Apple, so you can omit it.

If an error trap is used (ON ERROR GOTO), type ONERR GOTO for the Apple. The functions are the same on both machines. The TRS-80's ERL can be simulated by PEEK(218) + 256* PEEK(219). ERL calculates the line in which an error occurred. Similarly, use PEEK(222) for Level II's ERR/2 + 1 to obtain the code number for an error. But beware! The error codes are different for each system.

RESUME is nearly the same in both Basics. In a TRS-80 program it's often followed by a line number. Put GOTO in the Apple error routine to handle that situation. RESUME NEXT on the TRS-80 has no equivalent in Applesoft.

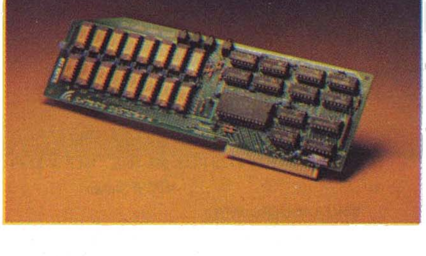
Poke has already been discussed, as it applies to output to the screen. Any other Pokes in a Level II program are probably to TRS-80 pointers, and may be nearly impossible to duplicate on the Apple. Similarly, Peek's three common functions are not easily simulated either. Peeks to the keyboard (14336 to 15359) have been discussed.

Peeking screen memory locations is impossible in Applesoft because



"What if ... I run out of memory?"

Most people *do* run out of memory with only 18K VisiCalc* workspace. But you can expand your Apple II* to 177K VisiCalc memory! You can also get 80-column display, lower case letters, and hard disk support—all without buying a bigger computer.



The Saturn expansion system for VisiCalc consists of a preboot diskette, one or more plug-in RAM boards, and an optional 80-column display board. You can put the Saturn boards in any slot. And with all that memory, our software lets you save files on more than one diskette.

Each Saturn RAM board includes additional software for other programming applications. So your BASIC, PASCAL, and CP/M programs get an extra bonus.

Ask your computer dealer for more details about the Saturn memory expansion systems. See how much bigger and better your models can become!

*VisiCalc is a registered trademark of VisiCorp. Apple II is a registered trademark of Apple Computers, Inc.

SATURN SYSTEMS INC.

P.O. Box 8050
3990 Varsity Drive
Ann Arbor, MI 48107

1 (313) 973-8422

Circle 81 on Reader Service card.

Program listing 5.
VARPTR simulation.

```
50 N1 = ASC(LEFT$(N$,1)):N2 = 0:R$ = RIGHT$(N$,1)
55 IF R$ = "%" THEN N1 = N1 + 128
60 IF LEN(N$)<2 OR (LEN(N$)<3 AND (R$ = "%" OR R$ = "$")) THEN 70
65 N2 = ASC(MID$(N$,2,1))
70 IF R$ = "%" OR R$ = "$" THEN N2 = N2 + 128
75 AD = PEEK(105) + 256*PEEK(106):EE = PEEK(107) + 256*PEEK(108)
80 FOR I = AD TO EE STEP 7
85 IF PEEK(I) = N1 AND PEEK(I + 1) = N2 THEN 95
90 NEXT I:PRINT CHR$(7):H8% = 0:M$ = " THAT VARIABLE HASN'T BEEN
ASSIGNED YET.":CALL 4200:RETURN
95 K = I + 2:RETURN
```

we're not using the Apple text screen. This capability can, however, be simulated by establishing an array which matches the TRS-80 screen memory to keep track of the locations of values or characters. To establish the array, DIM SC%(63, 15) then assign any ASCII value from the TRS-80 screen to the matching locations in the array. The top right corner, for example, if assigned the character A, would be SC%(63,0) = 65. A program could then check for an A in the top right corner (IF SC%(63,0) = 65 THEN. . .).

Peeks to any other location are specific to the TRS-80 and aren't easily transferable. Any machine-language subroutines, such as music or fast graphics that are poked in by a TRS-80 program, must be entirely rewritten from Z-80 to 6502 code, or else scrapped. If you are not fluent in assembly language, avoid TRS-80 programs with machine language, or at least that particular part of the program.

Simulating VARPTR(A$)

VARPTR(A$) stands for VARiable PoinTeR, and it is a difficult statement to implement on the Apple. It locates the starting address in memory of any variable. Any program that uses it is probably doing some fancy stepping.

The variable name in the parentheses can be located in the Apple's variable table that begins at the address held in memory locations 105 and 106 in page zero. Peek these locations, then scan the variable table to find the variable's correct name. This memory location stores the value for a numerical variable or the length and address of a string variable.

Each variable occupies seven bytes in the Apple's variable table. The first two bytes store the name. Integers have 128 added to the ASCII value of each of the characters in the name. The integer sign (%) is not stored. So, AA% would have 128 added to the ASCII value for each A (65) stored in each location, making 193 193. Following the two name bytes comes the high byte of the variable's value, the low byte, and three zeros to fill the extra spaces.

A single precision number is stored with both letters' ASCII value intact, so AA would be stored as 65 65. The

TRS-80	APPLE-80
INPUT "Name";N$	H8% = xxx:M$ = "ΛNAME?":CALL 4200: GOSUB 9:N$ = B$
INPUT N	H8% = xxx:M$ = "?":CALL 4200:GOSUB 9: N = VAL(B$)
K$ = INKEY$ (this doesn't check for a key actually being pushed)	K8 = PEEK(– 16384):K$ = " ":IF K8>127 THEN K$ = CHR$(K8 – 128):POKE – 16368,0 GET K$
205 K$ = INKEY$:IF K$ = " " THEN 205 (this does wait for a key press)	
IF PEEK(14400) = 32 THEN 2000 (i.e. if the left arrow is depressed GOTO that routine at line 2000.)	IF PEEK(– 16384) = 136 THEN 2000

Table 3.
Input routines compared.

next byte is the exponent, followed by four bytes for the mantissa of the value.

A string value has 128 added only to the second character of the name. Thus, AA$ would be 65 193. The next byte is the string length. The memory address of the string is stored in the next two bytes, low byte first and high byte second. Two zeros fill out the remaining spaces.

Only the first two characters of any variable's name are stored, and a single letter name has the second byte either 0 or 0 + 128. For example, A$ would be 65 128.

Using this information, K = VARPTR(A$) on the TRS-80 can be simulated by N$ = "A$":GOSUB 50 which calls the Applesoft subroutine in Program listing 6. In place of the A$, put whatever variable name is in the parentheses following the VARPTR statement in the TRS-80 program.

One important caution. The Apple stores variable values highest byte first and lowest byte last. If you converted these values from a TRS-80 program, reverse their order in the Apple program. Consult both manuals if you must delve into this particular area. String addresses, on the other hand, are stored identically by both machines. Since the most frequent use of VARPTR is for what's called string packing, problems should be rare.

USR(N) is almost the same on each machine. For the TRS-80, poke the machine-language routine's address into 16526 and 16527. The Apple uses 11 and 12. (You must also POKE 10,76 on the Apple.) For example, 150 POKE 16526,J:POKE 16527,K:N = USR(2) can be duplicated by 150 POKE 10,76:POKE 11,J:POKE 12,K: N = USR(2). The value in parentheses is passed to the routine. Any program which includes USR uses a machine-code subroutine which has to be completely rewritten.

LLIST and LPRINT send TRS-80 output to the printer. The Apple uses PR#1, followed by the normal list or print commands, or whatever instruction your computer happens to

Table 4.
Apple and TRS-80 Peek addresses.

Key	TRS-80 PEEK(N) where N =	TRS-80 value returned from PEEK(N)	Apple PEEK(– 16384) returns
@	14337	1	192
A	14337	2	193
B	14337	4	194
C	14337	8	195
D	14337	16	196
E	14337	32	197
F	14337	16	198
G	14337	128	199
H	14338	1	200
I	14338	2	201
J	14338	4	202
K	14338	8	203
L	14338	16	204
M	14338	32	205
N	14338	64	206
O	14338	128	207
P	14340	1	208
Q	14340	2	209
R	14340	4	210
S	14340	8	211
T	14340	16	212
U	14340	32	213
V	14340	64	214
W	14340	128	215
X	14344	1	216
Y	14344	2	217
Z	14344	4	218
0	14352	1	176
1 !	14352	2	177/161
2 "	14352	4	178/162
3 #	14352	8	179/163
4 $	14352	16	180/164
5 %	14352	32	181/165
6 &	14352	64	182/166
7 '	14352	128	183/167
8 (	14368	1	184/168
9)	14368	2	185/169
* :	14368	4	170/186
+ ;	14368	8	171/187
<	14368	16	188/172
= –	14368	32	189/173
>.	14368	64	190/174
? /	14368	128	191/175
ENTER (RETURN)	14400	1	141
CLEAR	14400	2	no equivalent
BREAK	14400	4	no equivalent
↑ (Λ)	14400	8	222
↓	14400	16	no equivalent
←	14400	32	136
→	14400	64	149
SPACE	14400	128	160
SHIFT	14464	1	does not register

require.

Four minor differences will probably never cause you any trouble, unless I fail to mention them.

First, the logic in If-Then statements and logic operations evaluates to minus one on the TRS-80, but to plus one on the Apple. Second, the Then in an If-Then statement is optional on the TRS-80, but *not* in Applesoft. Thirdly, the TRS-80 prints numbers with a space preceding (unless that space is used for a negative sign) and another space following the number. The Apple includes no such spaces around its numbers. Fourth, an Applesoft time delay loop counts about three times as fast as the TRS-80 loop. Increase any values in such loops appropriately.

Statements You Can Omit

Some TRS-80 statements need not be converted because they aren't used in Applesoft. CLEAR is a perfect example. On the TRS-80, CLEARn sets aside a specified amount of string space. The Apple, however, automatically allots all available extra memory for strings. (Incidentally, Applesoft has its own Clear statement, but this one nulls variables. Don't confuse the two!)

Similarly, FRE(A$) on the TRS-80 calculates how much string space remains unused. In Applesoft, FRE(0) calculates how much memory is unused (and remains available to strings). Applesoft's FRE(0) also substitutes for the TRS-80s MEM function.

RANDOM reseeds the TRS-80's random number generator. This is not a critical function, and may be omitted.

CINT and CSNG force the result of a calculation to become an integer or single precision number, respectively. As discussed for DEFINT and DEFSNG, Applesoft will do this by appending the % sign for integer, and using no special character for a single precision variable.

Statements That Must Be Omitted

The five remaining TRS-80 statements are machine specific and can't be duplicated on the Apple without enormous effort. Avoid INPUT#, PRINT#, PRINT USING, IN, and OUT. Additionally, any variable with the # sign after its name, such as K#, is double precision, a capability Applesoft doesn't provide. Similarly, DEFDBL and CDBL can't be converted without writing your own double precision routines. CHR$(23) creates double width text on the TRS-80 screen. The Apple can't duplicate that easily, but I have used SCALE = 2 to draw double size (both width and height) characters to approximate (and sometimes improve upon) the effect.

CHR$(N), where 0<N<31, are control codes of various types. Examples are backspace (8), linefeed (26), or clear to the bottom of the screen (31). These functions would need special routines to implement them.

A Sample Program

Program listings 3 and 4 compare a sample TRS-80-to-Apple conversion. The original TRS-80 program draws a simple bar graph from input data. The

Apple conversion duplicates the program exactly. The converted program was easy to translate without altering message length, graphics or layout.

In the Apple version, line 0 must be the first program line. It established M$ as the first, and H8% as the second, variable in the Apple's table. The machine-language routines used these positions to locate strings to print and to fetch their screen positions, and thus are vital to the converted program's success.

Line 1 defines the disk command character (control-D), and then relocates the Basic program above hires screen 1, if the program's not already there. This makes room from memory locations 2048 to 8192 (6K) for the shape table and machine-language routines. This also leaves 22K from 16384 to the start of DOS for the converted program (on a 48K Apple). This doesn't mean, however, that a 22K TRS-80 program will fit into this area. The Apple conversion requires more memory since it must do things in a more complex manner. Nevertheless, any converted 16K TRS-80 program ought to be smaller than 22K, and to fit easily.

Line 2 dispatches some preliminaries and then loads the binary file containing both the shape table and the machine language that accesses it. Line 5 is the subroutine that clears the screen, while lines 9 to 29 simulate the Input command.

Line 600 of the TRS-80 listing prevents the cursor from reappearing on the screen and erasing part of the graphics. Since the Apple cursor doesn't appear on the hires screen, the Applesoft program can use END in its place.

The remainder of the program can be compared line by line, to see how the conversion goes. Quite straightforward. You don't even need to see the original program run on a TRS-80 to recreate it on your Apple.

Several enhancements to existing TRS-80 programs are now possible. Using the Apple's own graphics, such as underlining text, or adding sound, good TRS-80 programs may be made better Apple-80 programs.

Here's one useful tip if you're using a color television as a monitor. Detune it slightly so that the display is in black and white. TRS-80 programs have no color, and black and white text is much clearer.

Any worthwhile TRS-80 program deserves the time spent to convert it for your Apple using this relatively simple process. The number of programs available to your Apple will dramatically expand. You need no longer envy TRS-80 owners their range of material, or pass up that TRS-80 magazine article or book that contains exactly the program you've been hunting for. Make it an Apple-80 program, and enjoy. ■

HUNTINGTON COMPUTING

Softlights

By Fred Huntington

Did you know we send out twenty thousand or more sales bulletins every two months?

Did you know that our sales are probably the best sales you've ever seen? Last month we sold the Elephants at an introductory price, the people who got the sales bulletins were able to buy them a month before the rest of the country.

One of the items in the current sale is ribbon cartridges (not just refills) for the MX-80 at **$4.99** each, MX-100 **$14.99.**

How do you get on the mailing list? There's only one way. That's to place an order. Requesting a catalog won't do it. But, once you buy from us, you're stuck on our mailing list.

Future sale flyers will contain fabulous bargains, both old and new items.

So, to entice you to want to get on the mailing list, there are some super specials listed elsewhere in this page.

SUPER-DUPER JOYSTICKS

Last month Barb ordered me a Wico joystick for evaluation without telling me first. When it came in I couldn't believe it. It is the most substantial joystick for the Atari I've ever seen. It's made by the same people who make the ones for the commercial arcade games. Made for the Atari, they will work with the Apple with an adaptor or the Joyport. There's also a fancier model with a chrome shaft and red ball on top. They also make a trackball controller that will blow your mind.

#160 **Command Controller** ($29.95) now **$24.44**
#161 **Red Ball** ($34.95) now **$29.44**
#162 **Trackball** ($69.95) now **$54.44**

These prices are good through Jan. 30, 1983.

VERBATIM CLEANING KIT - $5.00

Here's a loss-leader that we hope will entice you to buy from us. When you make any purchase, you can but a $12.95 Verbatim cleaning kit for only $5.00. One to a customer or family, only while supply lasts. When this ad was written, there were 400 on the way to us. So, when these are gone, that's it. Order stock #4012.

GENERIC COMPUTER GAMES

We're revolting! Whoops, I mean we're revolting against high prices in computer games. As part of our effort, we are publishing the only Generic Computer Games in the world. Originally costing $14.95 each, I've taken Skeet Shoot, Trap Shoot (both HIRES) and put them on one disk along with a picture of Great Grandma Huntington.

They are two of the first games ever produced by a very famous software company. I bought the rights for them for $200 and decided to have some fun with them.

Made for the Apple, they come complete with a yellow documentation page, reminiscent of what you would find in your local supermarket. (After all, our shipping warehouse was at one time a supermarket.) Order #9008 for only **$9.99.**

WURST OF HUNTINGTON

We're still getting rave notices about the Wurst of Huntington. People are amazed at how we cut the price of Wurst from $19.99 to **$9.99.** Once they see the program, however, they understand.

One customer wrote us saying that the Wurst was a tribute to beginning programmers all over the world who would like to sell their programs. Another praised us saying Wurst was everything we promised.

So, if you want the worst eight programs ever assembled on one disk (and have a little fun, too), order #9010 for **$9.99.**

COMPUTER ALMANAC

We think that Computer Almanac, written by Dave Carman is a good buy. It's a weather predicter, a loan amortization program, and a wealth of general knowledge. Order #9014 for **$24.99.**

UNDERSTAND YOURSELF

Written by Mike Taylor, Understand Yourself is a series of tests to help you get a better psychological insight of yourself. Not meant to be a clinical substitute, Understand Yourself is fascinating and will keep you and your mate intrigued for hours. Order #9012 for **$24.99.**

Here's something new for all you Epson MX-80 and MX-100 lovers — the MXPLUS. This amazing gadget allows you to use your printer buttons to instantly select the print mode you want.

For instance, press the "on line" button three times and you have instant double size print. Press twice and you have condensed print. No soldering, cutting or drilling. Easy to install. Does not interfere with normal operation of printer or software. Our price **$44.99.**



Great Grandma Huntington Cared

INTEGER FIX, ETC.

Here's an easy-to-use program that will covert your Integer programs to Applesoft in nothing flat. #9543. **$19.99.**

We also carry the complete line of Market software including Tennis Draw, League Schedule, Long Term Reservations, Swim Meet, League Standings and more. Call for prices.

We also have Masterworks, Nutricalc, Execuware, PCP, Thesis, and much, much more.

Here's a special to catch before the price increase. We're selling Advance Access disk boxes ($24.95 list before price increase) for **$21.99.** Buy two or more and you can have them for **$19.99** each. These are super storage boxes and will hold up to 90 disks each. I have five of them at home.

We've got Pac-Man Ghost pins for $2.50, Super-Fan II for $59.00, Master Diagnostics Plus - for $55.00, Mail Ordering program $5.00, and Verbatim disks 10/$25.99 with a plastic carrying case.

STREET LIFE

We now carry Street Life, as featured in Play Boy. We don't recommend this for everyone and will not sell it to you if we even suspect you are under the age of twenty-one. Warning, some people may find the theme and language offensive. Our price, **$26.99.**

ATARI ATARI ATARI ATARI ATARI ATARI

Our collection of Atari games continues to grow. My personal favorite now is Frogger. Until Jan. 30 you can have this $34.95 game for only **$19.94** when you purchase any other item. Please specify cassette or disk. The graphics and sound on this program are superb.

Airstrike by English Software is a new game that has been very popular overseas. It only takes 16K and comes in cassette and disk versions. If you're bored by ordinary arcade-type games, Airstrike will present an enormous challenge to you. List price is $39.95. Until Jan. 30 you can have it for **$29.94.**

THANK YOU

Thank you to the many well-wishers on the birth of our son, Dale. He is very healthy and doing well. He's learning the computer business from the ground up. He's by Barb's desk every day. I'm sure many of you have heard him in the background when you call. You should see Barb trying to take an order over the telephone, punch numbers into the computer and nurse Dale, all at the same time.

One last item, we have a spiffy new lower case chip that's made for us locally. For a short time we are selling it for only **$15.00.** I wanted to call it the Buffalo Chip but I got out-voted. So, for $15.00 be sure to order the GGH lower case chip. (Can you guess what GGH stands for?)

HUNTINGTON COMPUTING
Post Office Box 1297
Corcoran, California 93212

Foreign Orders 209-992-4481
In California 800-692-4146

Apple® is a registered trademark of Apple Computer, Inc.
Pet® is a registered trademark of Commodore.
TRS-80® is a registered trademark of Tandy Corp.
Atari® is a registered trademark of Atari, Inc.

Outside Calif. 800-344-5106

We take **MasterCard, American Express or VISA** (Include card # and expiration date). California residents add 6% tax. Include $2.00 for postage. Foreign and hardware extra. Foreign (excluding Canada): remit U.S. currency, checks on U.S. banks, use listed charge cards, or make direct wire transfers through Security Pacific Bank, Corcoran, for a $6.00 charge. All overseas orders shipped by air. Send for free catalog. Prices subject to change without notice.

Circle 37 on Reader Service card.

Hexism Revealed

A BITter pill to swallow,
A bigger BYTE to chew,
Along comes hexadecimal
Just in time for you!

by Paul Raymer

Having spent a considerable time reading the many computer magazines available to the home and business computer owner, I realized that most of the articles and programs described in these periodicals relate specifically to hobby, entertainment or business activity.

It's apparent that the use of decimal numbers is rapidly decreasing. Basic programs are limited in scope and the dawning of hexadecimal is fast approaching. Fractions, of course, are a lost art.

Most articles seem to deal with the day-to-day problems of computer use; that is, learning the language, applications and use in home, industry and school.

It seems that the whole impact of computer use may very well be missed. The use of the computer in schools, offices and at home will change society's way of doing things.

Look how the home video recorder ruined popcorn sales at many porno theatres!

The impact that is yet to come will surely be in the language we speak and write. No longer will ordinary methods of communication be acceptable. When the computer finally makes the big step beyond—when the average college student no longer wishes to merely play Pac-Man-type arcade games, when the family doctor no longer finds it fun to balance her checkbook with the use of her computer, when school kids are fed up with Dungeons and Dragons and VisiCalc retails for $25 so it may gain some national acceptance, then the educators and the magazine editors will realize the greatest potential of the computer for social change. . . hex!

This delightful subset of decimal mathematics, too long the exclusive domain of assembly- and machine-lan-

George Washington, our No. $0001 President, was born $0002/$0016/$06C4 in Virginia. His early childhood was spent on a farm. His father died when he was $000B. He studied mathematics and when he was $0010 went to live with his half brother Lawrence, who built and named Mt. Vernon.

Lawrence died in $06D8 and George acquired his property by inheritance. He valued land, and when he died he owned $0046 $03E8 acres in Virginia and $0028 $03E8 acres in what is now West Virginia.

He married Martha Dandridge Custis in $06DF.

He was made commander-in-chief by the Continental Congress on $0006/$000F/$06EF.

He was resourceful, a stern disciplinarian and the $0001 strong, dependable source $0004 unity. He was elected president and inaugurated $0004/$001E/$06FD on the balcony of New York's Federal Hall.

He was reelected $0700 but refused to consider being president $0003 terms.

George Washington died $000C/$000E/$0707.

Political Science Example. George Washington.

1. How many railroads on a Monopoly board? $0004
2. How much do you collect for passing Go? $00C8
3. How much is luxury tax? $004B
4. What is rent on Boardwalk with one hotel? $07D0
5. How much does it cost to buy the Electric Company? $0096

Game/Quiz Example. The game of Monopoly, a quiz with answers.

Paul Raymer is chief programmer for Paul's Electric Computer (PO Box 42831, Las Vegas, NV 89116), Nevada's seventh-largest computer company.

guage programmers and others of that ilk, will certainly become an important part of our lives. It will be better understood by those who now take the computer for granted and consider it only a gadget, a toy, just an educational device or merely a business tool.

There's no need to explain this system or to compare it to the decimal system any more than we would want to use Roman numerals in this modern day and age. It should be enough to present some examples of how hex numbers will replace the non-computerized system now used.

The most astute will quickly understand and grasp these fundamentals and realize the potential of hex in fields of astronomy, geography, history, political science, religion, commerce, mathematics, and particularly education and computer sciences. Realistically, it may prove of little value in television, bartending and jogging.

The examples which follow have been prepared using an Apple computer, which only understands hex as $0000. The Radio Shack computers use 000H, and there are certain to be other variations. It is hoped that one day all of these prefixes and suffixes, with their inherent sexist overtones, may be done away with. Hex can then be treated as it is entitled to be, simply as 0000.

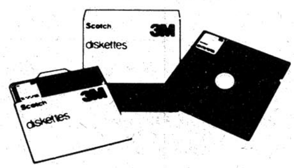
The author, writer of the book *A Hex on You All*, welcomes your comments. Please send a postage-paid self-addressed envelope for a crayon-written reply. ■

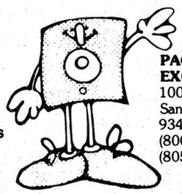
Sing a song of $0006pence,
A pocket full of rye;
$0004 and $0014 blackbirds
Baked in a $0003.$0588!

Children's Example. Nursery rhyme.

—————————	Chicago	Honolulu	New York	San Francisco
CHICAGO	————	$10A0	$02CA	$0743
HONOLULU	$10A0	————	$1369	$095E
NEW YORK	$02CA	$1369	————	$0A0C
SAN FRANCISCO	$0743	$095E	$0A0C	————

Commercial Example. Airline distances between cities.





The Power

Write programs that will run on over 40 different brands of mini and microcomputers? Absolutely correct, thanks to the UCSD p-System.

by R. John Buczek

How many programmers, hacker or professional, can claim to have written programs that will run on over 40 different brands and models of mini and microcomputers, with absolutely no modification?

I can. But don't send my super-hacker pin yet—I'm not alone. I'm just one of many progressive programmers using the UCSD p-System. Of course I was clever enough to get in on the bottom floor, more than two years ago, but a few notable companies have followed in my footsteps. You know the kind of company I mean—small progressive outfits such as Apple, Corvus, Hewlett-Packard, Philips and Texas Instruments, all of which have major commitments to the UCSD p-System.

Why p-System?

The advantages seem so clear that I have trouble understanding the lack of general acceptance. Of course, many people don't know much about the p-System, and there's a lot to know. Many have a big investment in other languages and operating systems, and are reluctant to change. But remember, Detroit is in big trouble in part because their industry didn't recognize the proper time to adjust to new market conditions.

Some people tell me they don't use the UCSD p-System because they don't like Pascal—an inappropriate response since the p-System is not just Pascal, any more than Unix is just C, or CP/M is just CBasic. The UCSD p-System is a complete operating system with several languages, many advanced features in common with the sophisticated operating systems of the mini and mainframe world, and very few features in common with the clumsy and primitive disk operating systems typically found in the micro world.

Elementary Features

The p-System includes the operating system, an editor, a file handler, a linker, libraries, a generalized assembler, three compilers and a number of utilities—a lot for your money, especially since it works so well.

I will be referring to Version 4.0, which is available through Softech and from some original equipment manufacturers. Several of the OEMs that jumped on the bandwagon early, such as Apple, Radio Shack and North Star, have arrangements directly with the regents of the University of California. These companies have versions 2.1 or earlier; they provide user support themselves, and their systems lack most of the new features. In most cases, however, the new versions are available directly from Softech even

R. John Buczek (PO Box 893, La Grande, OR 97850) met a computer for the first time in 1965 at the University of Denver and has hardly been out of reach of one since. After a hitch with a R&D unit in the Navy, more school at the University of Washington, and four years at a medical research lab, he has settled down in the Oregon high country. Following several years of selling, servicing and programming micros, John is now organizing his own "electronic cottage industry" by doing off-site testing and verification for software houses all over the country. He would be happy to answer your comments or inquiries.

Apple Corporation
Commodore
Data General Nova
Digital Equipment Corporation
Heath/Zenith Data Systems
Hewlett-Packard
IBM Personal Computer
Lockheed Sue
Nanodata QM-1
NCR Alp II
North Star
Ohio Scientific
Philips P2000
Radio Shack
Sage 68000
Sperry Univac
Texas Instruments
Western Digital MicroEngine
Xerox 820
VAX

Table 1. Compatible hardware.

Table 2.
Commands.

Adjust: Use cursor control keys to move an entire line across the screen, center a line in the middle of the defined line.
Copy: Copy a portion of text from: the buffer which always contains the last text inserted or deleted, or from another file entirely or in part, or from markers previously placed in the current file, to the current location of the cursor.
Delete: Use cursor control keys, space bar or return to delete text. Deleted text is retained in the buffer until the next insert or delete operation.
Find: Find the first or the indicated occurrence of a specified word or sequence of letters.
Insert: Accept anything keyed in as text except certain control characters.
Jump: Move cursor/screen point of view from current location to the beginning or end of the file, or to a previously declared marker, or to the point where the last insert, delete, copy, etc. occurred.
Margin: Reformat a paragraph in accordance with new left, right, and paragraph indent data.
Quit: Quit by: Exit to system. Save under existing name. Write to a new name. Update as a system work file. Return to the same document. Change to another document.
Replace: Replace a given target word, string, paragraph, etc., with a specified substitution. Do this 1, 2. . . n times as specified. If desired, require verification before each replacement. If the two strings are not the same length, then reformat in accordance with current margin specifications.
Set: Set a marker, by name, to permit location of specific areas in a long document; or set a pointer to mark the beginning of a copy, delete, etc.; or set the environment to specify margins, automatic indentation, automatic carriage returns, and certain other parameters.
Exchange: Replace text on a character by character basis so as not to disturb existing formatting, etc.
Zap: Delete large areas of text.

though the hardware people won't tell you about it.

Many other manufacturers (see Table 1) have current licenses with Softech and can provide the latest versions. If your machine is not directly supported, but the processor is, Softech offers the Adaptable System version. This comes with complete directions for connecting the system with your special combination of drives, and so on.

How It Works

The system is based on the concept of a pseudocode interpreter, a full definition of which is a book in itself. Programming is compiled, not into the machine code of a particular CPU, but into a standard optimized code, called a pseudo machine code. This code does not execute directly on any standard processor, but is executed by running a p-machine, a machine-language interpreter that must be written for each different CPU in which the system will run. Writing this p-machine typically requires only a small fraction of the effort that would be required to create an entirely new system for each machine.

Let's run over the key points again: Each program is compiled into p-code, which is exactly the same no matter what computer produced it, then executed by the p-code interpreter, which is different for every processor. A true compiler would produce native machine code, which would execute faster (typically two or three times faster), but with the penalty that native code files are much larger, and thus consume more RAM and disk space.

An ordinary interpreter, as in most varieties of Basic, results in programs that run 30–50 times slower than p-code, with very large code files that are in fact modified text files. This inflicts an even bigger hardware penalty, since nothing is optimum except ease of program modification, the primary excuse for Basic's continued existence.

The operating system is complete. It permits multitasking with asynchronous processes and 128 levels of priority, program chaining and runtime I/O redirection. The system's segmentation feature allows a 64K machine to run programs with code files many times larger than 64K, automatically calling each part of the program as it is needed.

Other advanced features include dynamic overlays, dynamic memory allocation, runtime support routines and block I/O service routines. Most impressive is the ability to accept code files compiled on another CPU. In one extreme case a very large program, intended for full-screen formatting, was composed on an Apple II editor (half-screen), then downloaded (uploaded?) to a PDP 11/34 minicomputer (due to memory and display limitations in this case). Finally the compiled code was downloaded to an Altos 8000-2, where it ran perfectly. Other systems offer portability, but they at least require recompilation.

The gossip underground has hinted about expansion of the multitasking primitives into a full-fledged multiuser system for some time. Softech's roots are in the mini world and I can't believe they would ignore the new generation of powerful processors, so this idea has credibility, although a high quality distributed network operating system is a good bet, too.

The screen-oriented editor comes with the system and has most of the features common to the current generation of word processors. Table 2 is a list of commands with a brief explanation of each. Cut and paste is no problem, and copying from files of standard paragraphs, program routines or formulas is easy. In implementations with sufficient disk space, the editor can automatically maintain a backup copy of each text file for protection. Most mistakes can be cancelled with a single keystroke. You can use this editor for all your correspondence, documentation and programming.

Did you notice that I said programming? The assembler and all the compilers accept text files from this editor or any other UCSD-compatible editor as input. Imagine using a word processor for Basic programming. Any program output can be diverted to a text file and reviewed or modified with the editor before printing.

The file handler permits you to do just about anything with a file you

couldn't do under your old operating system. As in all of the p-System, every option is prompted across the top of the screen, so you don't need to run for the manual to look up the command for the feature that you use only occasionally. If you have a clock in the hardware, the system knows it and automatically labels every file with the date and time of the last access. If you don't have a clock you can enter the date and time manually.

The file handler will transfer any kind of file from one device to another, and list a simple directory of any storage device or an extended directory that also lists type of file, size of file and starting location of the physical device.

You can check the status of any hardware device, rename a disk or file, crunch files together to condense available space or even modify the directory itself. An excellent feature is the ability to scan a disk for bad blocks of data, determine if the problem is physical damage on the surface of the disk, and order the system to avoid that area permanently if there is physical damage.

If the disk surface checks out OK, the system will try to correct the bad data, at least to the point of restoring directory pointers and block organization so that you can reenter the damaged file and salvage the remaining data. Here in the mountains of Oregon, power surges and frequent interruptions make this capability precious.

Another related feature is the option to maintain a duplicate directory on each disk and utilities to restore the original should it bomb.

The Linker

One linkage method is to order the system to copy a precompiled portion of programming into the current program, thus limiting recompilation to the parts that have been modified, with obvious increases in programmer productivity.

Another approach is to leave parts of the code in a run-time library and require the program to look it up and load it from that library every time the program is run. With this feature you can have an entire disk full of unrelated programs, but only have to store (and write) the I/O, screen handling, graphics and other general purpose routines once. Neither the linker nor the system is sensitive to whether or not the various units to be linked were originally in the same language. A number of vendors sell UCSD "tool kits" of fancy database, graphic and other useful routines that can be called from any of the languages supported.

The libraries are intimate partners of the linker. The System Library comes partially loaded with routines that require special handling on your machine. These routines are supplied by the vendor, and most vendors provide enough information to permit modification by someone with the necessary expertise. You can add additional code units, standardizing various operations by a number of different programs, or conserving space if the code must appear in every program. I'll talk later about how this can assist marketing. Versions before 4.0 permitted only the System Library to be active, but the newest release permits several specialized libraries to be active at the same time.

The UCSD Adaptable Assembler is a one-pass generalized assembler, with a hardware-independent core, and special modules to handle the various CPUs supported. Once you've learned it you need only (only??) learn the new machine codes to change machines, rather than learn an entirely new editor/assembler.

Most of the features of the most advanced assemblers are provided, including macros, linking, conditional assembly, parameter passing, and so on. Some special features of the UCSD system are easy linkage into high-level programming, parameter passing from high-level programming and easy accessibility of system parameters. Either fixed or relocatable code can be specified.

I haven't done any serious assembly-language programming since school, but this assembler would have taken most of the pain out of my early exposure. You can write a series of assembly-language routines for various machines, put them into a single library, and have a high-level program that can run on different machines and will call only the versions appropriate to

those machines.

For the advanced programmer and forward-looking software house, cross assemblers spanning any of the supported processors are available. These currently include: Z8, Z-80, 8080, PDP/11, LSI 11, 6502, 6800, 6809 and 9900, 8086, and 68000, with other 16-bit processors likely. Each processor has a complete set of cross assemblers so that any machine can produce machine code for any other.

Pascal Is Prime

The system currently supports three high-level languages: Pascal, Fortran and Basic, all compiled. Much has been written about Pascal. I've used most of the common languages, and I find that the discipline, precision and rigor that Pascal enforces greatly improves not only the end product but also the skills and productivity of the programmer.

I have worked with a number of young programmers trained at our local state college and have found in every case that programmers who learned Pascal first perform far better than other programmers.

Some people say they find Pascal restrictive, but the UCSD system permits a suitably skilled programmer to disable most of the restrictions. UCSD Pascal conforms closely to the proposed ANSII standard Pascal, with a number of extensions that are picked up in new Pascal compilers from other vendors. UCSD Fortran is in fact ANSI-77 standard Fortran with most of the features of the full language.

A Better Basic

The Basic compiler is an advanced structured Basic offering all the usual features and a number of useful extensions such as the if. . . then. . . else conditional and virtual arrays. I have seen several business programs originally written in Microsoft Basic or CBasic translated into UCSD Basic without much trouble (providing the number of peeks and pokes are held to a minimum). After translation they ran faster.

In all three languages you can program on the system editor. The advanced features of the operating system, such as multitasking, external

SETUP: Customizes the system to get the most out of your specific hardware by modifying 36 system parameters, including control keys, line delay, disk size, etc.

BINDER: Modifies the X-Y screen addressing routines for different terminals.

SCREENTEST: Run this after SETUP or BINDER to make sure that all screen handling routines are correct.

LIBRARIAN: Creates and modifies libraries.

BOOTER: Copies the bootstrap onto a new disk.

PATCH: Inspects or modifies any 512-byte block on a disk.

COPYDUPDIR: Restore a bombed directory if the duplicate directory option is active.

MARKDUPDIR: Toggle the duplicate directory option.

DISASSEMBLER: This program reads a UCSD code file and outputs symbolic pseudo-assembly code text along with various statistics about the program. A debugging tool for advanced programmers.

Table 3.
Softech utilities.

units, libraries of standard routines, and so on, are available to all languages. The system has arithmetic operations to 36 digits of precision.

Native Code Speed

The system took a giant step forward when Softech recently announced their native code compilers, to complement the p-code compilers. Now time-critical programs, or portions of programs, can be compiled into native code for each machine. The key feature of the system is still the p-code portability, but native code compilation can provide extra speed.

What Next?

I asked Softech about future compilers, but they declined to discuss any new products until they are ready for market. One representative did say that many new products are under development which may include new compilers. Rumors about C and Ada compilers have been circulating for some time. It is widely known that Professor Ken Bowles, father of the UCSD system, has taken an extended leave of absence to work on an Ada compiler.

Utilities provided vary with the vendor, some charging extra for all the Softech utilities and some adding more of their own. On any machine I've used, the Turtlegraphics routines are better than the graphics routines supplied by the hardware OEM. Softech's latest utility is Xenofile, which gives you access to CP/M data files. Table 3 lists the standard Softech utilities.

Other utilities are available from various vendors, from Softech, or from USUS, the UCSD user's group, if you become a member.

All documentation is professional, complete and readable. Manuals, especially those from Softech, are printed on good stock and well bound. Organization and indexing is good. Tutorial instruction manuals are available, as are formal handbooks. If you get everything, the stack of manuals is at least a foot high.

Business Concerns

Applications end users (business users in case you didn't recognize yourselves) may be wondering what all this will do for them. Their goal, as everyone's, is better, more complete, cheaper programs. Here are some ways the p-System can help all of us.

The primary factor affecting software price is the size of the potential market. If I am writing a program on a make and model of computer that has only 10,000 units installed, that is the absolute limit to my market. Only if I am willing to assume the headache and overhead of supporting different versions, or if the language dialect and operating system are exactly duplicated on other machines, can I exceed

"The advantages I've discussed are portability, security, speed and compactness."

this limit.

The first option usually isn't practical and the second just doesn't happen. Even a supposedly standard combination, CP/M and Microsoft Basic for example, requires a depressing amount of fine tuning for each machine unless all I/O operations are minimal, with no graphics, no clever screen handling, no sound, and, in general, nothing that couldn't be done with a Teletype 38.

The p-System is a better solution—it is the most portable system in the industry. Nothing is more portable, not for micros, minis or mainframes, and the UCSD system will run on at least some machines in each class. My guess is that writing in UCSD opens the door to a potential market of around 500,000 machines. This means bigger markets, more money and lower unit prices for everyone. Hurrah!

Another problem facing the marketing world is just how much source code access an end user has a right to demand, balanced against how much the company can risk handing out. If the source code is available, even a beginning programmer can file off the serial numbers, do some cosmetic work and offer it as his own. The pirate cuts the price and frequently sells more copies than the originator. Easy to see why suppliers are reluctant to supply source code.

Even in Basic, some houses are going to extraordinary lengths to make the code unintelligible and hard to modify—unfortunately, easy modification is the characteristic the Basic user pays for with slow execution and large code and data files.

Some vendors are using modified operating systems that render a disk uncopyable as well as unmodifiable. This may be all right for games but it is entirely and completely unacceptable for business software that is important to the day-to-day operation of the company. The uncopyable VisiCalc sold lots of copies and helped to sell lots of Apples, but with the arrival of a number of excellent copyable look-alikes, it's goodbye Apple-VisiCalc, hello CP/M-SuperCalc.

One reason for the continuing soft sales of the Apple III is Apple's hard-wiring of this scheme into the SOS Operating System. Hard disks are becoming more affordable all the time, and a major reason for buying them is to get that stack of 20 different program disks off a busy executive's desk. If a machine won't conveniently boot a program from the hard disk, the business-person might buy something else.

The capabilities of microcomputers are increasing to the point that many former mini users are taking a serious look at micros, but buying an application program without the source code is almost unknown in that market.

A final source code consideration is the specter of failure in our industry. More than half of the companies entering this market in the last four years are already out of business, often having left their customers lost and alone. Many of our OEMs have recently begun major marketing efforts aimed at the Fortune 500 or similar groups, and you can bet that they know about that statistic. These businesses want to be assured of continued support.

A related problem is the difficulty that noncomputerists have in evaluating a program before they buy it. Most suppliers will not refund a purchase if the customer is later dissatisfied, and customization is often a real necessity. About 25 percent of the companies buying their first piece of business software run up against major weaknesses in their application—without source code they often have to write it off as a total loss. Word gets around, and this makes end users leery.

First-time buyers will often accept programs without the source code, but second-timers are reluctant and third-time buyers usually adamantly refuse. We cannot continue to stuff square companies into the round holes of a single general business package that cannot be customized.

Many trade associations in various industries (medicine, for example) are advising their members never to buy a business program unless they get the source code. First-time buyers that don't know a byte from a baud often are up on this point, and consider it unnegotiable.

Something must be done to satisfy the legitimate concerns of both sides. Without some security scheme, marketing people may double or triple the price of a program, to recover their development cost before piracy destroys the market.

My solution, as you might suppose, is the p-System. And Table 4, which lists software vendors who are into Pascal, shows I'm not alone. The critical parts of a program, the unique routines that I originate as the bulk of my creative process, I write as subroutine calls or external units that I compile separately and put into a library. I then write what I call the frame of the program separately. The frame includes, in addition to the outermost layer of subroutine calls, menus, accesses to data files, most input formatting and all output formatting.

On request I give the source code for the frame to the end user. With this, any competent programmer can reformat, link in his own routines or modify the basic flow of the program. This satisfies most of the typical requests for modification.

All they need to know about my special routines are the subroutine names and passing parameters, which give them access while giving no critical programming away. Thus, I'm protected, and the customer doesn't feel nearly as helpless.

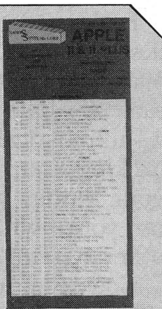
There are a number of nearly undetectable ways to put a unique identification on each copy of the library routines, so I can always identify the original owner of any pirated copies. I include cash penalties for permitting such copies in the licensing agreement, thus eliminating the need to prove damages in court. Court action is then limited to enforcing a mutually agreed-upon contractual penalty. Where appropriate I include in the library units a copyright slug that will print out on the screen.

In one case the customer rebuilt a program to the point that I hardly recognized it. He was able to make the program more useful to himself, while my safeguards still protected me. The new native code compilers should permit more variations on this theme.

Compact Files

Unlike most dialects of Basic, which store all data as ASCII text, UCSD offers automatic binary packing of data. Some of my customers saved the cost of

inCider BOOK SHELF



NANOS CARDS

Nanos System REFERENCE CARDS FOR APPLE II, APPLE II PLUS, MODELS I, II, III AND COLOR COMPUTER

At last! No more flipping through the pages of the BASIC manual! No more working through the maze of machine language instructions! These cards completely summarize the BASIC and Assembler manuals! FEATURES INCLUDE:
memory map, eyeball graphics, math instructions, BASIC commands, store instructions, BASIC functions, load instructions, BASIC statements, move instructions, special keys, exchange instructions, PRINT USING examples, shift instructions, BASIC special characters, compare instructions, BASIC and assembler messages and codes, branch instructions, BASIC facts, data alteration instructions, reserved words, I/O instructions, ROM routines, complete character chart with graphics and space-compression codes, hex-dec chart, control code cross-reference, assembler instructions, commands and operators, screen line layout, editor commands and subcommands, condition code easy access.
Plus—"magic graphics number—a mystery until you learn how to use it!
Designed as a fold-up, accordion-style card, fits in your pocket. Panels organized for optimum speed for reference.

Apple II and Apple II Plus: BASIC and 6502 FC1008	$4.95
Apple II and Apple II Plus: BASIC only FC1007	$3.95
Model I: BASIC and Assembler FC1002	$4.95
Model II: BASIC and Assembler FC1005	$5.95
Model II: Commands and Utilities FC1010	$3.95
Model III: BASIC and Assembler FC1005	$5.95
Color: BASIC and Extended FC1006	$4.95
Z-80: Microprocessor FC1011	$4.95
ZX80, ZX81 and Timex Sinclair 1000: FC1012	$5.95
Pocket Computer: BASIC FC1009	$2.95

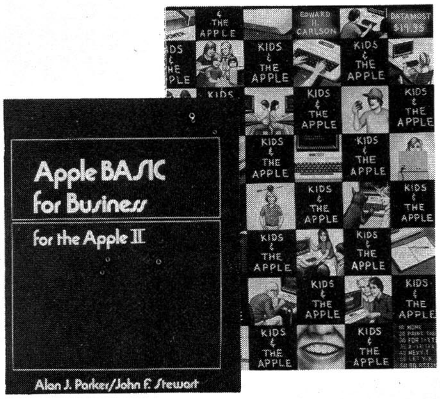
THE APPLE II USER'S GUIDE—By Lon Poole, Martin McNiff, and Steven Cook This guide is the key to unlocking the full power of your Apple II or Apple II Plus. Topics include: "Applesoft and Integer BASIC Programming"—especially how to make the best use of Apple's sound, color and graphics capabilities. "Machine Level Programming," "Hardware Features"—which covers the disk drive and printer, and "Advanced Programming"—describing high resolution graphics techniques and other advanced applications. Well organized and easy to use. BK1220 $16.95

NEW

APPLE BASIC FOR BUSINESS: for the Apple II—by Alan J. Parker and John Stewart. Unlike most introductory BASIC books, this book uses files extensively. It is written specifically for the Apple II microcomputer with DOS Version 3.2. All programs presented are compatible with DOS Version 3.3. With the emphasis on problem-solving, the focus of this book is the point at which problem elements meet language capabilities. BK1247 $15.95

ASSEMBLY LANGUAGE PROGRAMMING FOR THE APPLE II—by Robert Mottolz. This comprehensive, easy to understand introduction provides solid groundwork for getting started in assembly language programming on the Apple II.* Many subroutines written in assembly language are provided, and most explanations are shown with equivalent examples in BASIC. There's an excellent section on hexadecimal arithmetic included, as well as appendices for further study. BK1249 $12.95

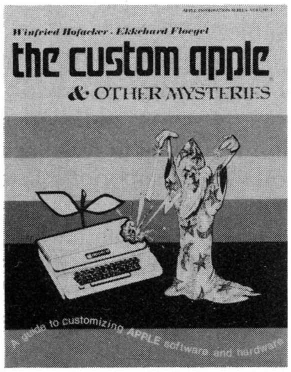
APPLE MACHINE LANGUAGE—by Don Inman and Kurt Inman. APPLE MACHINE LANGUAGE builds upon your previous knowledge of BASIC, and teaches you the machine language in small, easy, completely illustrated steps. Following this guide, you will be able to write machine language programs directly, using the Apple System Monitor. Each new program is thoroughly presented in functional blocks, with sketches of how each step will actually appear on the video screen. Soon you will be entering and executing your own machine language programs, with predictable results! BK1248 $14.95



MOSTLY BASIC: APPLICATIONS FOR YOUR APPLE II, BOOK 1—by Howard Berenbon. This book provides you with 28 ready-to-use, BASIC language programs which have been completely tested and debugged for use on your Apple II. Includes a telephone dialer, digital stop watch, spelling test, a house buying guide, a gas mileage calculator, and many others useful to businessmen, hobbyists, scientists, and computer enthusiasts. BK1251 $12.95

MOSTLY BASIC: APPLICATIONS FOR YOUR APPLE II, BOOK 2—by Howard Berenbon. A second gold mine of fascinating BASIC programs, including two dungeons that test your math and history abilities and another one that's strictly for fun, eleven household programs, a monthly savings plan and six more on money or investment, two that test your level of ESP, and more—32 in all! Excellent for beginning or advanced computerists. BK1252 $12.95

KIDS AND THE APPLE—by Edward H. Carlson. Whether you are a kid, a parent, or a teacher, this book is something unique. It starts with the bare bones introduction to programming, leads quickly to more interesting programs, and gives anyone who uses it a complete knowledge of Applesoft BASIC. Lively illustrations, notes to parents and teachers and questions for the reader are sprinkled throughout the book. While this guide is aimed at 8-16 year olds, adults will find it equally attractive as a beginning book for use the Apple personal computer! BK1253 $19.95



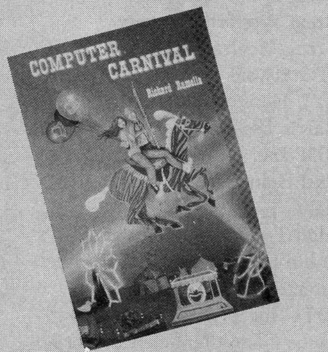
SOME COMMON BASIC PROGRAMS, APPLE II EDITION—By Lon Poole et al. A powerful collection of financial, statistical, home management and mathematics programs—76 in all. Each program is presented with BASIC source code, operating instructions and descriptions. If you're a beginning programmer you can learn from this book what well designed and documented programs look like. BK1232 $14.99

THE CUSTOM APPLE AND OTHER MYSTERIES—by Winifred Hofacker and Ekkehard Floegel. This is *the* guide to customizing Apple software and hardware, published by the folks at IJG. It contains such hands-on information such as: data acquisition and control applications, Programming the 6522 Internal timer, Constructing the 6522 I/O board, An Eprom Burner for the Apple Computer, An Eprom/RAM board, The Apple Slot Repeater, and much, much more. BK1246 $24.95.

FOR YOUR OTHER SYSTEM

COMPUTER CARNIVAL—by Richard Ramella. Your child can become a crackerjack computerist with the sixty TRS-80 Level II programs in COMPUTER CARNIVAL. This large-type, spiral bound book for beginners is a veritable funhouse of games, graphics, quizzes and puzzles. Written by *80 Micro* columnist Richard Ramella, the programs are challenging enough to ensure continued learning, yet short enough to provide your child with the immediate delight and reward of mastering basic computing skills. And for even greater enjoyment, get the CARNIVAL COMPANION, a 30-minute cassette containing all the programs in the book. Eliminates tiresome typing and lets your child spend more time enjoying the programs. BK7389 $16.97 CC7389 Book and Cassette $24.97

THE SELECTRIC INTERFACE—by George Young. You need the quality print that a daisy wheel printer provides but the thought of buying one makes your wallet wilt. SELECTRIC™ INTERFACE, a step-by-step guide to interfacing an IBM Selectric I/O Writer to your microcomputer, will give you that quality at a fraction of the price. George Young, co-author of *Kilobaud Microcomputing* magazine's popular "Kilobaud Klassroom" series, offers a low-cost alternative to buying a daisy wheel printer. SELECTRIC INTERFACE includes: step-by-step instructions, tips on purchasing a used Selectric, information on various Selectric models, including the 2740, 2980, and Dura 1041, driver software for Z80, 8080, and 6502 chips, tips on interfacing techniques. With SELECTRIC INTERFACE and some background in electronics, you can have a high-quality, low-cost, letter-quality printer. Petals not included. BK7388 (125 pages) $12.97



TEXTEDIT—A Complete Word Processing System in Kit Form—by Irwin Rappaport. TEXTEDIT is an inexpensive word processor that you can adapt to suit your differing needs—from form letters to lengthy texts. Written in TRS-80 Disk BASIC, the system consists of several modules, permitting the loading and use of only those portions needed. A disk is also available which provides the direct loading of the modules, however, the book is required for documentation. For Model I and III with TRSDOS CONVERT., one disk drive (2 disk drives or copy utility needed to transfer to system disk). Runs under TRSDOS 2.2/2.3. May not function under other systems. BK7387 $9.97 Disk DS7387 $19.97

For Toll Free Ordering Call 1-800-258-5473

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *inCider* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.50 for the first book, $1.00 each additional book for U.S. and foreign surface. $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

Table 4.
UCSD p-System users.

Aardvark Software Inc.
Addison-Wesley Publishing
Advanced Business Technology Inc.
Advanced Computer Design
Apparat Inc.
Apple Computer Inc.
Applied Software Technology
Arizona Computer Systems Inc.
Arkansas Systems Inc.
Automated Management Services Inc.
Automated Medical Office Systems Inc.
Beaman Porter Inc.
Blue Lakes Software
Broderbund Software
Business & Professional Software Inc.
Business Planning Systems
Business Solutions Inc.
BUS Pascal Software
Cavri Systems Inc.
Chapin Associates
Columbia Computer Associates
Computer Options
Computer Task Group
Comshare Target Software
Context Management Systems Inc.
Cybermedic Associates
The Datalex Company
Datamed Research Inc.
DB Master Associates
Denver Software Company
Duosoft Corporation
Ecotope
Efficient Management Systems Inc.
Energy Data Systems Inc.
Escape Ltd.
Ferox Microsystems
Franklin Institute Research Lab Inc.
Paul Friday
James Gagne
Ginger Bred Software
Graphic Software Systems Inc.
Great Plains Computers
Great Western Software
Gregg/McGraw-Hill Publishing
ST Software
H. J. Hansen Company
Harris Technical Systems
HDP Inc.
Carl Helmers
Helu
High Technology Software
Robin Hill Systems Ltd.
Interactive Technology Inc.
IOTC Inc.
Irvine Computer Sciences Corp.
Paul L. Juell
Learning Tools Inc.
Link Systems
MedComp Inc.
Merrimack Systems
Microfinancial Corp.
MicroPro International
MINDex Infosystems
Morgan-Fairchild Graphics
North American Technology Inc.
Organic Software
Osborne/McGraw-Hill
Pacific Data Systems
Pascal & Associates
Pascal Systems Inc.
PCD Systems Inc.
P-E Consulting Group
Pegasus Systems
Powersoft Inc.
Professional Business Software
Professional Data Services Inc.
Relational Systems International Corp.
Sage Software Company
Scenic Computer Software Corp.
Shakti Systems Inc.
Silicon Valley Software Inc.
Sirius Software Inc.
Sir-tech Inc.
Softsel
Software Consulting Services
Software Express
Software Institute
Software Products International
Software Publishing Corp.
Software Sorcery Inc.
The Sorites Group Inc.
Southwestern Data Systems
Statcom Corp.
State of the Art Inc.
Stoneware Inc.
Sunrise Software Inc.
Synergistic Systems Software
T&W Systems Inc.
Terak Corp.
Ticom Systems Inc.
Timberline Systems Inc.
VisiCorp Personal Software
Volition Systems
Western Software Development
C. J. Wigglesworth
XIPHIAS

Overseas

Advanced Technology Devices Inc. (Philippines)
Blyth Computers Limited (England)
J-Ph Carillon (France)
Cox, Lawrence and Associates (Australia)
Dynamic Control Systems (Canada)
Equinox Computer Services (Australia)
Tom Gilb (Norway)
Inria-Kayak (France)
Intelligence (England)
ITT Consumer Products Ltd. (United Kingdom)
Japan Business Automation Co. Ltd. (Japan)
Shi Microcomputers Ltd. (Israel)
Shattock and Associates (Australia)
Six "S" Business Advisory Pty. Ltd. (Australia)
Socite LERTIE (France)

a hard disk and backup by using UCSD programs that required 30–50 percent less disk capacity than similar programs in Basic. This almost halved the system hardware cost for these business people.

Packing can be done with many Basics, but requires extensive custom packing and unpacking routines for each new data file—the p-System does it automatically. If the automatic packing and data typing are not to your liking, you can disable it and use your own scheme, just as in any other system.

Code files are more compact than either interpreted or compiled languages. I have sold copies of a sophisticated General Ledger that has over 120,000 lines of source code, and yet the compiled code fits easily on one Apple format disk.

The Choice Is Yours

The advantages I've discussed are portability, security, speed and compactness. There are others. There are some disadvantages as well, not the least of which is dealing with Softech. Cost cannot be overlooked. The entire package carries a pretty hefty price tag, but OEMs can buy a license for a run-time kernel from Softech at a reasonable price per unit. This allows the end user to choose to buy the rest only if he requires it. From the programmer's point of view, the completeness of the system makes the price acceptable.

A good argument can be made for any of the alternatives available today—no operating system/language combination would be in use if many people didn't find it superior for some purpose. Nonetheless, something must be done to help us remain afloat in the face of wildly escalating software costs, constant demand for more sophistication and constantly increasing competition.

If we don't solve our quality control and compatibility problems, the new generation of hardware may prematurely destroy many good companies, only because that generation is more focused on these problems. The UCSD p-System is my solution. It could be that little extra advantage that pushes you up to the top of the stack. Even if your next programming effort isn't quite the VisiCalc or Pac-Man of tomorrow, the vastly larger market that the p-System can provide may bring you enough cash to keep going until you hit the right combination. ■

– MORE –
on
CRACK-SHOT

- ★ CRACK-SHOT is easy to use
  Designed for Amateur and Professional
- ★ Automatic B-file maker
  B-file can run on another APPLE
  without the card installed.
  B-file may be copied COPY A
  B-file may be sent over Modems
- ★ Fantastic Utilities
- ★ Huge Documentation
- ★ HOT-LINE 24 hours/day 7 days/week
  for any system upgrades thru
  PIRATES HARBOR
  (Over 1000 Users Now) $15.00 Extra
- ★ Great Product Support
- ★ CRACK-SHOT hardware contains 8 chips & 2 switches including E-prom and add on memory. Use in any slot other than 6.

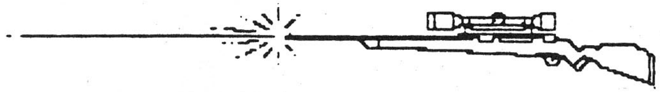
For Mastercard or Visa Only

Call Now! TOLL FREE

1-800-824-7888
Ask for Operator 68

in CA 1-800-852-7777
Ask for Operator 68

CRACK-SHOT



THE ULTIMATE IN COPY DE-PROTECTION

Not just another bit copier.

Blows Locksmith and Nibbles Away — AWAY

Even cracks all bit copiers in fifteen seconds

CRACK-SHOT is a hardware device that fits right in an Apple Slot and dumps all memory to disk.

- ★ Use as a gaming tool to stop, start, and save a game at any level.
- ★ Full monitor capabilities to examine, modify, trace single-step, or dis-assemble an interrupted program
- ★ Copies from DOS 3.2 to DOS 3.3
- ★ Faster and easier to use than any bit copier
- ★ Lets you suspend work with one program while you use another
- ★ Cracks single disk access programs in fifteen seconds, longer for multi-access

System Requirements: 48K Apple II or Apple II Plus
1 Disk Drive
Ramcard helpful but not required

Complete with Software and Instructions $149.95
Cash, Check, Money Order, Mastercard of Visa accepted
Order now from:

PIRATES HARBOR
P.O. Box 8928, Boston, MA 02114 617/738-5051 MODEM

Apple is a registered trademark of Apple Computer, Pirates Harbor is the Trademark of the Red Rebel
Locksmith—trademark of Omega Microware, Inc. Nibbles Away—trademark of Computer: applications

Crack-shot has been designed to be used as a gaming tool, programmers aid, and as a device to assist you in legal archival copies of your programs.
Crack-shot should not be used for illegal purposes.

Circle 76 on Reader Service card.

Apple Data— A Bumper Crop

Here is the first of a three-part series that explains all you'll ever need to know about speeding up your file searches. No strings attached. . .

by Peggy Burnett

Searching for a particular record in a file can make your computer operate at a snail's pace, causing increased frustration instead of the increased productivity you'd like. This series of articles presents some techniques to speed up the process. This month's installment explores file access methods for sequential files, and future installments will cover access methods in direct-access files and indexed files. Table 1 summarizes the techniques explained this month.

First, let me explain a few conventions. The term "sequential file" describes a file whose entries are organized just like a string of beads, one bead right after the previous one. Examples of this are a file whose entries are ordered by employee's social security number or arranged alphabetically by customer name, and a file with each entry in the order in which it was entered into the computer.

I'll describe ways to find a particular record in such a file, and present an algorithm, or "pseudo-program," for each. The algorithms are not written in any particular programming language. Instead they are written less formally, with the emphasis on logic flow rather than syntax. You should have no difficulty translating the algorithms to your own programming language.

Peggy Burnett is a partner in the computer systems consulting firm of Bulgren and Burnett Inc., and specializes in small-computer applications. She holds an M.S. degree in computer science, and has been a data processing consultant to small businesses since 1972. Address correspondence to PO Box 1355, Lawrence, KS 66044.

The Unsorted Sequential File Search

This is the slowest access method of all, but also the easiest to program. I include this method mostly for comparison with the other methods, since most of you are already familiar with it. You can add to the file by writing records sequentially after the end of the file, and retrieve a record by starting at the beginning of the file and reading sequentially until you find the right one. This method is suitable for small files, and can be used on tape as well as disk files. Listing 1 shows the simple algorithm to do this type of search.

Accesses: You would have to read an average of half of all the records in the file to find a particular one using this method. In a file containing 1000 records, this means an average of 500 accesses. If the one you're looking for isn't in the file, you'll have to read all 1000 records before you find that out.

Advantages: This method is easy to program, and takes no storage in addition to the actual data in the main file.

Disadvantages: It is very slow.

The Sorted Sequential File

The only difference between this kind of file and the unsorted sequential file is that the file is physically arranged "in order." For example, a sorted sequential file of employees is ordered by employee number or social security number, arranged alphabetically by name, or put in some other logical sequence. This allows you to tell whether your current position in the file is before or after the position of the record you're looking for. You can use this extra piece of knowledge in several ways.

The sequential search takes the same number of accesses as in an unsorted file to find a record that does indeed exist in the file (500 accesses in the 1000-record file I mentioned earlier). However, you can also determine in an average of 500 accesses whether the record you need is in the file, as opposed to having to read all 1000 records in an unsorted file. Listing 2a shows the algorithm for this kind of search. This method could be done just as easily on tape as on disk.

The binary search is much faster than the sequential search on disk files (you wouldn't use it on tape files), and is not difficult to program. You first check the middle record in the file to determine if its value is greater or less than the one you want. If it is greater,

Program listing 1. Algorithm to search an unsorted sequential file.

```
PROCEDURE PREPARE_TO_SRCH.
/* SETS UP PARAMETERS, THEN CALLS A SUBROUTINE TO ACTUALLY PERFORM */
/* THE SEARCH.                                                      */

FIRST_RECORD = 1.
LAST_RECORD = LAST_RECORD_IN_FILE.
INPUT "WHAT IS THE KEY VALUE OF THE ONE YOU WANT?", THE_ONE_WE_WANT.

CALL UNSORTED_SRCH (THE_ONE_WE_WANT, FOUND, FIRST_RECORD, LAST_RECORD).
END PREPARE_TO_SRCH.

PROCEDURE UNSORTED_SRCH (THE_ONE_WE_WANT, FOUND, FIRST_RECORD, LAST_RECORD).
/* IF THE_ONE_WE_WANT IS FOUND, RETURNS THE VALUE 'TRUE' IN THE
   VARIABLE 'FOUND'.  OTHERWISE, 'FOUND' IS SET TO 'FALSE'.  */

   RECORD_NUMBER = FIRST_RECORD.
   FOUND = FALSE.

   REPEAT
      READ RECORD_NUMBER RECORD.
      IF RECORD = THE_ONE_WE_WANT
         THEN FOUND = TRUE.
      RECORD_NUMBER = RECORD_NUMBER + 1.
   UNTIL FOUND OR RECORD_NUMBER > LAST_RECORD.

END UNSORTED_SRCH.
```

you know that the record you're looking for is in the first half of the file; otherwise it is in the second half. This simple check lets you eliminate half the records in your file with only one access! You then repeat the process by checking the middle record of the half you're still interested in, thereby eliminating half of those records in only one more access. And so on. This kind of search takes: $\log_2 (n) + 1$ accesses, where n is the number of records in the file. This works out to be a mere 11 in our 1000-record file example. The algorithm is shown in Listing 2b.

The interpolation search, also called the "telephone book" search, can be used in a sorted sequential file when you have a pretty good idea of where in the file your record is located. For example, in an alphabetical file of customers, you would expect to find Zelda Zimmerman near the end of the file. If you were looking up her name in the telephone book, you wouldn't look for her at the beginning (as in the sequential search), or at the middle (as in the binary search), but near the end. The interpolation search works the same way. After determining the distribution of data in your file, you can put a formula or a table into your program that tells you where a particular record (theoretically) should be found.

Here is an example. Suppose you have a 1000-record sorted file filled with account numbers between 1 and 9999. After doing a little analysis, you have learned the typical distribution of these account numbers in your file. Let's say they look like Table 2. For simplicity, assume that the numbers within each range are evenly distributed. So if you want account number 6050 (in the 6001–7000 range), it should be located 50/1000 of the way between 63.6% and 64.8% through the file, which figures out to be 63.66%. (I got the number 63.6% by adding 1.2% + 1.9% + .3% + ... + 49.6% from Table 2. Adding 1.2% more to 63.6% gives us 64.8%.) If you multiply 63.66% by 1000 (the number of records in the file), you learn that account number 6050 should be at record 637.

So far, so good. You know where the record *should* be. You then read record 637 to see if you guessed right. But suppose record 637 has account number 8000 in it. This account number is theoretically 99.4% of the way through the file (64.8% + 34.6%,

	Comments	Average Number of Accesses in a 1000-record File		Advantages	Disadvantages
		If record is in the file.	If record is not in the file.		
Sequential search in an unsorted file.	Also suitable for tape files.	500	1000	1. Easy to program. 2. No extra storage needed.	1. Very slow.
Sequential search in a sorted file.	Also suitable for tape files.	500	500	1. Easy to program. 2. No extra storage needed. 3. Better than the sequential search in an unsorted file, if the record is not in the file.	1. Slow. 2. File must be sorted whenever additions are made.
Binary search in a sorted file.		11	11	1. Not difficult to program. 2. Quite fast. 3. No extra storage needed.	1. File must be kept sorted.
Interpolation search in a sorted file.	Distribution of data in the file must be known. Principle can be adapted to some extent for tape.	5	5	1. Fastest of all methods in sequential files. 2. No extra storage needed.	1. File must be kept sorted.
Partitioned file (unsorted).		100*	200*	1. Easy to program. 2. File does not have to be sorted.	1. Extra storage may be needed.

*Assumes file is divided into 5 partitions of 200 records each.

Table 1. Summary of file access methods in sequential files.

```
PROCEDURE SORTED_SEQ_SRCH (THE_ONE_WE_WANT, FOUND).
/* FINDS THE RECORD IN THE FILE CORRESPONDING TO THE_ONE_WE_WANT,
   AND RETURNS THE VALUE 'TRUE' OR 'FALSE' IN THE VARIABLE
   'FOUND'.  */

   RESET FILE POINTER TO BEGINNING OF FILE.
   FOUND = FALSE.
   NOTFOUND = FALSE.

   REPEAT
      READ NEXT RECORD.
      IF RECORD = THE_ONE_WE_WANT
         THEN FOUND = TRUE
         ELSE IF RECORD > THE_ONE_WE_WANT
              THEN NOTFOUND = TRUE.
   UNTIL FOUND OR NOTFOUND OR END_OF_FILE.

END SORTED_SEQ_SRCH.
```

Program listing 2a.
Algorithm to search a sorted file sequentially.

```
PROCEDURE BINARY_SRCH (THE_ONE_WE_WANT, FOUND).
/* USES THE BINARY SEARCH TECHNIQUE TO FIND THE_ONE_WE_WANT IN
   THE FILE.  SETS 'FOUND' TO TRUE OR FALSE.  */

   LOWER = 1.
   UPPER = LAST_RECORD_IN_FILE.
   FOUND = FALSE.

   WHILE LOWER <= UPPER AND FOUND = FALSE DO
      MIDDLE = INTEGER ((LOWER + UPPER) / 2).
      READ MIDDLE RECORD.

      IF THE_ONE_WE_WANT > RECORD
         THEN LOWER = MIDDLE + 1.    /* NEXT TIME, TRY UPPER HALF */
      IF THE_ONE_WE_WANT < RECORD
         THEN UPPER = MIDDLE - 1.    /* NEXT TIME, TRY LOWER HALF */
      IF THE_ONE_WE_WANT = RECORD
         THEN FOUND = TRUE.
   END_WHILE.

END BINARY_SRCH.
```

***Program listing 2b.** Binary search algorithm.*

Account Number Range	Percent of Total
1-1000	1.2%
1001-2000	1.9
2001-3000	.3
3001-4000	10.0
4001-5000	.6
5001-6000	49.6
6001-7000	1.2
7001-8000	34.6
8001-9000	.4
9001-9999	.2
	100.0%

Table 2.
Distribution of the account numbers in the Interpolation Search example.

from our table again), so you need to figure out where to try next.

A handy little formula goes with this method that works nicely to tell you where to look next. The neat trick about this formula is that it works on any kind of distribution your file might have, because you use it on the percents (which are by definition uniformly distributed) rather than on the actual account numbers.

THIS_REC# = LO_REC# + (FRACTION* SECTIONSIZE) − 1
where: SECTIONSIZE = HI_REC# − LO_REC# + 1
and FRACTION = (THIS_% − LO_% + 1) /(HI_% − LO_% + 1).

So, in your first calculation, you had:

FRACTION = (63.66 − 1 + 1) / (100 − 1 + 1)
= .6366
SECTIONSIZE = 1000 − 1 + 1
= 1000
THIS_REC# = 1 + (.6366 * 1000) − 1
= 637.

For your second try, since account number 8000 is too high, you set:

HI_% = 99.4, and HI_REC# = 637

and compute your new values as follows:

FRACTION = (63.66 − 1 + 1) / (99.4 − 1 + 1)
= .64
SECTIONSIZE = 637 − 1 + 1
= 637
THIS_REC# = 1 + (.64 * 637) − 1
= 408

So you try record number 408 next. Suppose it contains account number 5000, which is supposed to be 14% of the way through the file. Since its value is too low, you set:

LO_% = 14 and LO_REC# = 408

and use the formula again:

FRACTION = (63.66 − 14 + 1) / (99.4 − 14 + 1)
= .59
SECTIONSIZE = 637 − 408 + 1
= 230
THIS_REC# = 408 + (.59 * 230) − 1
= 543

So the next try will be at record number 543. You repeat this process, narrowing the range each time as in the binary search, until you either find an entry or run out of records to try.

One caveat—you must know the distribution of your file to use this technique. If the distribution cannot be predicted or if it is unstable, you may lose efficiency rather than gain it using this method.

I have heard it claimed (but I can't prove it) that the average number of accesses using this method is: $\log_2 (\log_2 (n)) + 1$, which amounts to about 5 in a 1000-record file. Whether or not this formula is exactly correct, the method has been demonstrated to be quite efficient in files with known distributions, and can even be adapted to some extent for tape files. One example of the interpolation search is shown in Listing 2c.

Accesses: You can find a record in the sorted sequential file in an average of 500 accesses (in our 1000-record file), 11 accesses or (maybe) 5 accesses, depending on whether you use the sequential search, binary search or interpolation search, respectively.

Advantages: Obviously, searching this type of file can be faster than searching an unsorted file, and the programming to do it is not difficult. No additional storage is required other than the main file, and two of the methods described (the sequential search and the interpolation search) can be adapted for use on tape.

Disadvantages: The file must be kept sorted. This means that whenever you make an addition, the file must be sorted again. Since sorting takes up a fair amount of time and storage space,

Program listing 2c.
Interpolation, or "telephone book" search.

```
PROCEDURE INTERPOLATE (THE_ONE_WE_WANT, FOUND).
/* THIS PROCEDURE IS WRITTEN FOR A FILE (OF AT LEAST 2 RECORDS)
   WITH EVENLY DISTRIBUTED VALUES BETWEEN 1 AND 2000. */

   DEFINE FUNCTION INTERP_FCN (LO, THIS, HI)
                    = (THIS - LO + 1) / (HI - LO + 1).
   /* RETURNS THE PERCENT (IN DECIMAL FORM) OF THE DISTANCE
      THROUGH THE SECTION WHERE THIS RECORD SHOULD BE. */
   END_FUNCTION.

   FOUND = FALSE.
   DONE = FALSE.

   /* INITIALIZE SECTION BOUNDARIES */
   LOWER = 1.
   UPPER = LAST_RECORD_IN_FILE.
   LO_VALUE = 1.
   HI_VALUE = 2000.

   WHILE NOT DONE DO
      SECTION_SIZE = UPPER - LOWER + 1.
      PERCENT = INTERP_FCN (LO_VALUE, THE_ONE_WE_WANT, HI_VALUE).
      RECORD_NUMBER = LOWER + INTEGER (PERCENT * SECTION_SIZE) - 1.
      IF RECORD_NUMBER < LOWER
         THEN RECORD_NUMBER = LOWER.
      READ RECORD RECORD_NUMBER.

      IF RECORD = THE_ONE_WE_WANT
         THEN DO
            FOUND = TRUE.
            DONE = TRUE.

      ELSE IF SECTION_SIZE < 2
         THEN DONE = TRUE.   /* THE_ONE_WE_WANT IS NOT IN THE FILE */

      /* ELSE RESET SECTION BOUNDARIES AND PREPARE TO TRY AGAIN */
      ELSE IF RECORD < THE_ONE_WE_WANT
         THEN DO
            LOWER = RECORD_NUMBER.
            LO_VALUE = RECORD.   /* THIS RECORD */
         ELSE DO
            UPPER = RECORD_NUMBER.
            HI_VALUE = RECORD.   /* THIS RECORD */
      ENDIF.

   END_WHILE.

END INTERPOLATE.
```

the sorted sequential file organization is more suitable for relatively stable files than it is for files you frequently add to and delete from.

The Partitioned File

This is a crude but effective compromise between the unsorted sequential file and the sorted file. The idea is to use the unsorted file (to avoid sorting it all the time), but chop it into partitions, each of which contains a certain range of values, more or less along the lines of the sections of the interpolation search.

For example, you might choose to divide your file into five sections of equal size (200 records in each section, in the 1000-record example), with the first section covering the letters A–E, the second covering F–L, the third covering M–N (lots of people in our customer file have names starting with those two letters), the fourth with letters O–R and the last with letters S–Z. So customers whose names begin with the letters A–E go into the first partition whenever they are added, those with letters F–L into the second, and so on.

Then, to retrieve a record from the file, you can look at the first letter of the customer's name, and only search one-fifth of the file (using the unsorted file search), instead of the whole file. The algorithm is given in Listing 3.

Accesses: The average number of accesses to find a record will be half the partition size if the record is in the file, or all the partition size if the record is not in the file. If a 1000-record file is broken into five partitions, this amounts to 100 accesses to find a record that is in the file (or 200 if it is not).

Advantages: This is almost as easy to program as the unsorted file search algorithm, yet it cuts the number of accesses dramatically. Also, you don't have to continually sort your file.

```
PROCEDURE PARTITION_SRCH (THE_ONE_WE_WANT, FOUND).
/* THIS PROGRAM WORKS ON A 1000-RECORD FILE THAT HAS BEEN
   PARTITIONED INTO FIVE SECTIONS OF 200 RECORDS EACH.          */

   FOUND = FALSE.
   LETTER = THE_ONE_WE_WANT (1).  /* THE FIRST LETTER */

   IF LETTER <= 'E' THEN SECTION_NUMBER = 1
     ELSE IF LETTER <= 'L' THEN SECTION_NUMBER = 2
     ELSE IF LETTER <= 'N' THEN SECTION_NUMBER = 3
     ELSE IF LETTER <= 'R' THEN SECTION_NUMBER = 4
     ELSE SECTION_NUMBER = 5.

   LAST = SECTION_NUMBER * 200.
   FIRST = LAST - 199.
   CALL UNSORTED_SRCH (THE_ONE_WE_WANT, FOUND, FIRST, LAST).
                                             /* SEE LISTING 1 */

END PARTITION_SRCH.
```

***Program listing 3.** Algorithm to search an (unsorted) partitioned file.*

Disadvantages: This technique is slower than the binary search and interpolation search. Also, you have a potential problem with overflow—if you run out of space in a given partition, you then have to decide where to put the record. The easiest solutions to this problem are 1) to also establish an overflow partition (which will cut into your search efficiency somewhat), or 2) to make the partitions big enough in the first place. Both of these solutions require extra storage. A third alternative is to make each partition a separate file, an approach that doesn't use much extra disk storage, but is a little more difficult to program.

Conclusion

You can use these techniques exactly as presented, or tailor them to fit your own applications. These methods offer impressive improvements in search times, especially for relatively stable sorted sequential files. However, many applications have files that are more complex, larger, or simply not sequential in nature. Next month, Part II will talk about direct-access files, and Part III in this series will treat indexed files. ■

Instant Graphics

Picture yourself creating creditable charts, graphs or graphic designs in no time at all—all you need is an Apple Graphics Tablet.

by James A. Gupton Jr.

Creative graphics programs, though difficult to develop, provide one of the best methods of teaching computer programming. Creative graphics requires imagination, visualization, a firm understanding of coordinate plotting and a knowledge of a high-level language like Basic. You can add further complexity by incorporating color and motion into the graphic design. But be prepared to spend more time on graphics programs than you would on other programs, because you'll need to run and check each step of the plotting as you develop your design.

The Easy Way

On the other hand, the Apple Graphics Tablet lets anyone who can doodle on paper produce instant color graphics. Even more important, your Graphics Tablet enters your design into RAM for transfer to disk storage, so you can reuse it any time you wish. The Graphics Tablet does more than generate doodles—it is an excellent engineering tool for recording charts, graphs and even printed-circuit designs. Simply affix the chart or graph to the tablet's surface with tape, and trace the analog or digital recording with the stylus.

The Graphics Tablet is basically a digitizer with a resolution of 2794 points on the X axis and 2794 points on the Y axis (for a total of 7,806,436 points within an active area of 11 inches square). This produces a resolution of 0.003937 inches or 0.1 millimeter. Unfortunately, a spot this small cannot be reproduced on the screen of a typical monitor and is actually invisible to the human eye.

The basic structure of the Graphics Tablet is a series of magnetostrictive paths for the X and Y axes on a substrate beneath the working active area. A strain wave is transmitted along all wires simultaneously, and the movable stylus on the working surface senses the passing strain wave. The time delay between the initiation of the strain wave and the sense time of the stylus is computed to code the X and Y coordinates of the stylus.

The Graphics Tablet comes with a special interface (Apple No. 600-0149) with cable connections for the tablet and stylus. It is inserted into one of the accessory edgecard connectors across the rear of the main Apple circuit board. You can use any connector slot except 0 and 6 for the Graphics Tablet interface since slot 6 is generally used for your disk drive and slot 0 for your Applesoft card. (See the photo.) The Graphics Tablet is initiated by software on the Apple 5¼-inch disk. When you load the disk, it displays the Apple logo and Graphics Tablet notice; press the escape key to display the menu.

Considering the resolution capabilities of the Graphics Tablet, you might assume that your monitor could reproduce perfect circles that would be free of edge distortion. Unfortunately, this is not the case. First, few CRTs can produce a spot smaller than 0.02 inches in diameter; color CRTs limit spot size to 0.03 inches. And the basic Apple graphics resolution is only 48 horizontal lines and 48 vertical lines. Even with the Apple high-resolution graphics, the best resolution is only 280 horizontal lines to 192 vertical lines. At best, any circle will exhibit step-like edges.

Tips on Using the Graphics Tablet

Once you have initiated the Graphics Tablet and selected the form of graphics from the menu, you can use the stylus to pinpoint your working surface on the monitor display. Draw a diagonal line from each corner to the center of the working surface and note the grid at which the line first appears on the CRT screen. Use masking tape to line off the active surface that corresponds to the CRT display area. Then just place the point of the stylus on the Reset square across the top of the tablet to clear the screen.

To draw straight lines, guide your stylus tip with a plastic 30-60 degree or 45-90 degree triangle. Likewise, use

NIBBLE *EXPRESS* Vol. I

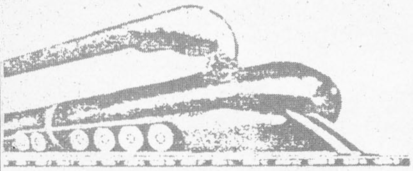
TABLE OF CONTENTS

SECOND PRINTING!

BRAND NEW PROGRAM LISTINGS!

nibble T.M.

EXPRESS! (for your Apple*)



ORDER NOW

All programs and Articles are centered on the Apple Computer family.

NIBBLE
P.O. Box 325
Lincoln, MA 01773

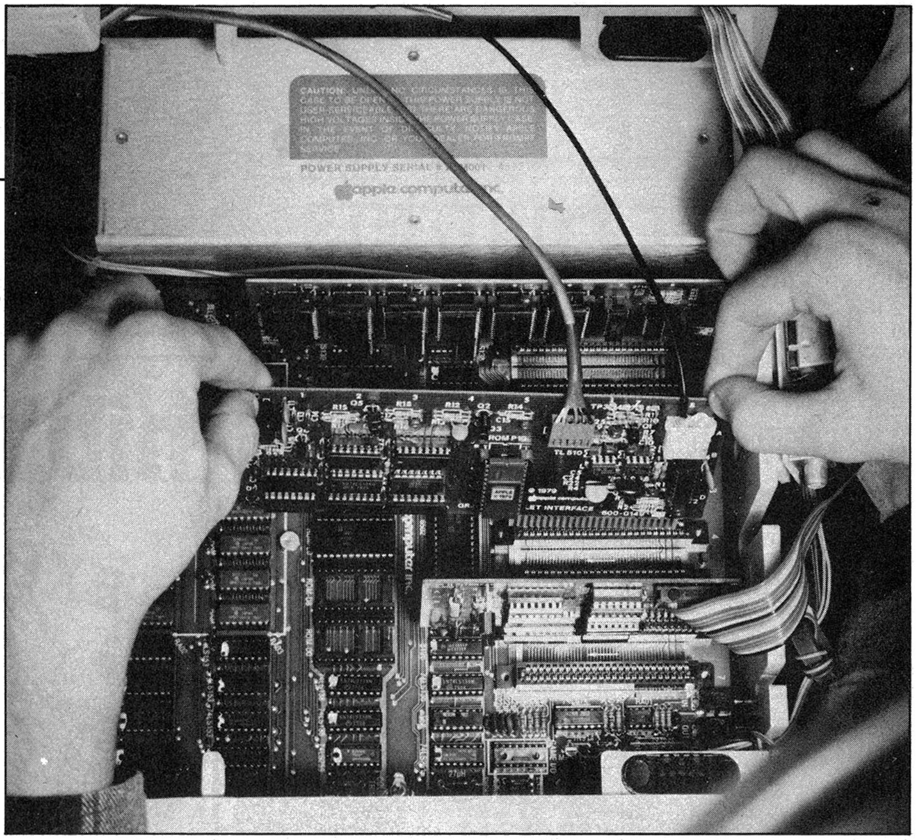
Yes! I want NIBBLE EXPRESS Vol. 1 in my library!
Here's my ☐ Check ☐ Money order
for $12.95 plus $1.75 postage/handling. (Outside U.S. add $2.75 Surface Mail or $5.00 Airmail.)

☐ Also send me NIBBLE EXPRESS Vol. 2 at $14.95 plus $1.75 postage/handling. (Outside U.S. add $2.75 Surface Mail or $5.00 Airmail.)

Master Card & Visa Accepted

Card # ____________ Expires ________
PLEASE PRINT CLEARLY
Signature ____________
Telephone ____________
Name ____________
Street ____________
City ________ State ____ Zip ____

Your check or money order must accompany your order to qualify.
Outside U.S.: Checks must be drawn on a U.S. Bank.
*Apple is a registered trademark of Apple Computer Company.



The Graphics Tablet interface can be installed in any available slot in the Apple.

plastic circle templates to draw circles. Never use a metal-edge rule or metal-tip compass on the working area of the tablet—metal drawing aids will magnetize the strain wave wires.

Draw color lines in the same manner as white lines; activate the desired color by pressing the stylus tip on the appropriate color square across the top of the tablet. To make a two-color line, draw the first line in the desired color, move the plastic line guide, activate the second color square with the stylus tip, and redraw the line.

The new multicolor graphics printers will print your Graphics Tablet designs in full color.

You can use the Graphics Tablet for children's games, art training, scientific investigations, and multicolor graphic analysis.

I won't claim that the Graphics Tablet can instantly make any Apple user a computer programming genius and precision line artist. But it will speed the learning process by eliminating the programming expertise that would otherwise be needed to reproduce graphics designs. ■

"The Inflation Fighter is Here"

The "ORANGE+™" $995.00

Suggested List Price



Finally, a dynamic powerful, practical personal computer that is fully Franklin Ace 1000 and Apple® II compatible, yet, lower in price than the competition. Uses existing available software. The "ORANGE+" is the pacesetter of the future, with a **switchable 110-220 volt power supply, upper and lower case with lower case locking key**, enhanced audio and color functions, 8 slot motherboard, game port, fully socketed I.C. board, a 3 ROM operating system, 48K ram, fully expandible.

The "ORANGE+" will interface with peripherals that will work on both the Franklin ACE 1000 or the Apple® II. Thousands of existing software programs, hardware, games, peripherals and accessories will plug right into the New "ORANGE+" COMPUTER.

Best of all is the Price.

The "ORANGE+" could easily be sold for $1500.00, But no, the inflation fighter retail price is only $995.00, thats right, Only $995.00.

Schools, Groups and Companies can now purchase a quality computer at a down to earth price. At these prices, now everyone can afford a powerful personal computer for work or play, add various peripherals and software to build a powerful computer that can do everything the competition can, but for a lot less money.

Ask your favorite dealer for the "ORANGE+" today, if he doesn't have one in stock tell him to write or call his local distributor now.

The "ORANGE+" is fully Apple II and Franklin ACE 1000 compatible.

The "ORANGE+" is fully warranteed for 90 days, with a 9 month extended warranty available for an additional $99.00.

Watch for future exciting products from the manufacturers of the "ORANGE+" COMPUTERS.

Selected Distributorships Available. Dealer Inquiries Invited.

For further information, contact Collins International Trading Corporation, 16311 Ventura Blvd., Suite 500, Encino, California 91436.

ORANGE+™

213-906-3776

Circle 14 on Reader Service card.

Franklin ACE is a trademark of Franklin Computer Corporation
Apple is a registered trademark of Apple Computer Inc.
Orange+ is a trademark of Collins International Trading Corporation



Perpetual Calendar

Need to know some significant fact about a particular date, Julian or otherwise? Then all you need do is enter and run this helpful program to get all the dates you'll ever need.

by Robert Suder

On what day was the Declaration of Independence signed? How many days have elapsed since man first landed on the moon, July 24, 1969? What is the corresponding Julian date for August 21, 1981? What day of the year was May 23, 1981?

You can find the answer to these questions using the program listing and your Apple II.

Background

The first calendar was probably based on the lunar month, that is, from one full moon to the next. This was not very satisfactory because the interval is 29.5 days—in just a few years, the months didn't correspond to the actual seasons. Then, in 46 B.C., Julius Caesar developed a more dependable instrument, but even the Julian Calendar was less than accurate.

The problem with inventing a calendar is that a solar year is not exactly 365 days; it's closer to 365.26 days. That leftover fraction of a day is why every fourth year is a leap year. Even though leap year is included, the Julian Calendar is imprecise, and throughout the years errors accumulated.

Finally, in 1582, Pope Gregory XIII decided to drop ten days in order to bring the calendar back in step with the seasons.

Now most of us use the Gregorian Calendar. All years exactly divisible by four are leap years, except the century years that are exactly divisible by 400. In 3000 years the Gregorian Calendar will be in error by less than one day.

Because of the complexities of the civil calendar, the Julian Day Calendar was invented. It simply counts the number of days that have elapsed since

Program listing. Applesoft Basic calendar.

```
10  REM    DAYS OF THE WEEK
20  REM    BY DR. ROBERT SUDER
30  REM  1 SEPTEMBER 1980
100  CLEAR
110 D$(0) = "SATURDAY"
120 D$(1) = "SUNDAY"
130 D$(2) = "MONDAY"
140 D$(3) = "TUESDAY"
150 D$(4) = "WEDNESDAY"
160 D$(5) = "THURSDAY"
170 D$(6) = "FRIDAY"
500  HOME : VTAB (5): PRINT "              TASKS AVAILABLE"
510  VTAB (8): PRINT "     1--DAY OF THE WEEK"
520  VTAB (10): PRINT "     2--DAYS BETWEEN DATES"
530  VTAB (12): PRINT "     3--DAY OF THE YEAR"
540  VTAB (14): PRINT "     4--JULIAN DAY"
550  VTAB (16): PRINT "     5--EXIT PROGRAM"
560  VTAB (20): PRINT "TYPE THE NUMBER OF YOUR CHOICE"
570  INPUT N
580  ON N GOTO 1000,2000,3000,4000,10000
590  GOTO 570
```

Listing continued.

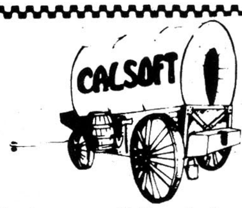
Dr. Robert Suder (5839 Downing, Portage, MI 49009) is employed by the science department in the Portage Public Schools.

apple discount software

CALSOFT

Personal—Entertainment—Business

SOFTWARE



We have all the latest software—ASK US!

Low Discount Prices / 15% to 25% Discount off List Price
Fast Convenient Service / We ship same or next day
Large Selection of Software / Call or Write for our FREE Catlog!

Call Toll Free: *(800) 423-5290* ***In California:*** *(213) 991-9641*

BEAGLE

	List Price	Our Price
DOS Boss	24.00	20.40
Utility City	29.50	25.05
Alpha Plot	39.50	33.55
Apple Mechanic	29.50	25.05

BRODERBUND

	List Price	Our Price
Apple Panic	29.95	25.45
David's Midnight Magic	34.95	29.70
★ **The Arcade Machine**	**59.95**	44.95
Star Blazer	31.95	27.15
★ **Choplifter**	**34.95**	26.20
Deadly Secrets	34.95	29.70
Serpentine	34.95	29.70
Sea Fox	29.95	25.45

HOME ACCOUNTANT
~~74.95~~ 56.20

CONTINENTAL

	List Price	Our Price
★ **CPA Modules #1–4 (each)**	**250.00**	187.50
First Class Mail	74.95	63.70

DATA MOST

	List Price	Our Price
Snack Attack	29.95	25.45
Swashbuckler	34.95	29.70
★ **Casino**	**39.95**	29.95
Pig Pen	29.95	25.45
Crazy Mazey	NEW .. 29.95	25.45
Mars Cars	NEW .. 29.95	25.45
Tubeway	NEW .. 34.95	29.70

EDU-WARE

	List Price	Our Price
Rendezvous	39.95	33.95
★ **Prisoner 2**	**32.95**	24.70
Algebra series (each)	39.95	33.95
Fractions/Decimals (each)	49.00	41.65
PSAT/SAT Word Skills (each)	49.00	41.65

HOWARD

	List Price	Our Price
★ **Tax Preparer (1983)**	**NEW 225.00**	168.75
Creative Financing	195.00	165.75
Real Estate Analyzer II	195.00	165.75

INFOCOM

	List Price	Our Price
Zork I, II, III (each)	39.95	33.95
★ **Deadline**	**49.95**	37.45
Starcross	NEW .. 39.95	33.95

MICROSOFT

	List Price	Our Price
Typing Tutor II	24.95	21.20
TASC Compiler	175.00	148.75
RAM Card	NEW .. 99.95	84.95
★ **Multiplan**	**NEW 275.00**	206.25

MUSE

	List Price	Our Price
Super Text 40/56/70	125.00	106.25
Castle Wolfenstein	29.95	25.45
Know Your Apple	NEW .. 34.95	29.70

PENGUIN

	List Price	Our Price
Complete Graphics System	69.95	59.45
Graphics Magician	59.95	50.95
Special Effects	39.95	33.95
Spy's Demise	NEW .. 29.95	25.45
Transylvania	NEW .. 34.95	29.70

SENSIBLE

	List Price	Our Price
Super Disk Copy III	30.00	25.50
Multi-Disk Catalog	25.00	21.25
★ **Sensible Speller**	**125.00**	93.75

SIERRA ON-LINE

	List Price	Our Price
HI-RES Adventures 0-4		15% OFF
★ **Time Zone**	**99.95**	74.95
Dark Crystal	NEW .. 39.95	33.95
★ **LISA**	**79.95**	59.95
LISA Educational System	119.95	101.95
★ **General Manager**	**229.95**	172.45
Threshold	39.95	33.95
★ **Screen Writer**	**129.95**	97.45
★ **Screen Writer Professional**	**199.95**	149.95
Cross Fire	29.95	25.45
Mouskattack	34.95	29.70
Ultima II	59.95	50.95
Frogger	34.95	29.70
Cannonball Blitz	34.95	29.70
The Artist	79.95	67.95
Laf-Pak	34.95	29.70
Pest Patrol	29.95	25.45
Lunar Leepers	29.95	25.45

FREE DISKETTE with each order over $50. Just mention this ad.

SIR-TECH

	List Price	Our Price
Wizardry	49.95	42.45
Knight of Diamonds	34.95	29.70
Star Maze	34.95	29.70

SOFTWARE PUBLISHING

	List Price	Our Price
★ **PFS**	**125.00**	93.75
PFS: Report	95.00	80.75
★ **PFS: Graph**	**125.00**	93.75

STONEWARE

	List Price	Our Price
DB Master	229.00	194.65
★ **DB Master Utility Pak #1**	**99.00**	74.25
★ **DB Master Utility Pak #2**	**99.00**	74.25

SYSTEMS PLUS

	List Price	Our Price
★ **General Ledger**	**395.00**	296.25
★ **Accounts Payable**	**395.00**	296.25
★ **Accounts Receivable**	**395.00**	296.25

TG PRODUCTS

	List Price	Our Price
★ **Joystick**	**59.95**	44.95
Game Paddles	39.95	33.95
★ **Select-a-Port**	**59.95**	44.95

SIRIUS SOFTWARE

	List Price	Our Price
Gorgon	39.95	33.95
Sneakers	29.95	25.45
The Joyport	49.95	42.45
Kabul Spy	34.95	29.70
★ **Bandits**	**34.95**	26.20
Escape from Rungistan	29.95	25.45
★ **Fly Wars**	**29.95**	22.45
Free Fall	29.95	25.45
Way Out	39.95	33.95
★ **Blade of Blackpoole**	**39.95**	29.95
Type Attack	39.95	33.95

WORD HANDLER
~~199.95~~ 149.95

STRATEGIC SIMULATIONS

	List Price	Our Price
Cartels & Cutthroats	39.95	33.95
Southern Command	59.95	50.95
★ **Napoleon's Campaigns**	**59.95**	44.95
Battle of Shiloh	39.95	33.95
Tigers in the Snow	39.95	33.95
Road to Gettysburg	59.95	50.95
Pursuit of the Graf Spee	59.95	50.95
Guadalcanal Campaign	59.95	50.95
Cytron Masters	39.95	33.95
Galactic Gladiators	39.95	33.95
★ **The Cosmic Balance**	**39.95**	33.95
SEUIS	39.95	33.95
★ **Germany: 1985**	**NEW . 59.95**	44.95
Battle of Normandy	NEW .. 39.95	33.95

VISICORP

	List Price	Our Price
★ **Visicalc**	**250.00**	193.75
Visitrend/Visiplot	300.00	255.00
Visidex	250.00	212.50
Visifile	250.00	212.50

We also carry complete lines from the following companies:

ADVENTURE INTERNATIONAL
ARTSCI • ASHTON-TATE
AUTOMATED • AVANT-GARDE
BEAGLE • CAVALIER
DATASOFT • DON'T ASK
GEBELLI • HAYDEN
INSOFT • KENSINGTON
LEARNING CO. • MICROLAB
MICROPRO • PEACHTREE
PHOENIX • PICCADILLY
QUALITY • SENTIENT
SORCIM • SOUTHWESTERN
SPINNAKER • SUBLOGIC
SYNERGISTIC • VIDEX
If you don't see it, Ask Us!

CALSOFT
346 N. Kanan Rd. #103
Agoura, CA 91301

Call Toll Free: *(800) 423-5290* ***In California:*** *(213) 991-9641*

We accept Mastercard & Visa (include # and Expiration Date), check, COD ($1.50 extra), or Money Order. California residents add 6% sales tax. Include $2.00 for shipping (UPS Blue Label $3.00, Canada $6.00, other foreign countries $10.00).

★ **Sale prices are through February only! Prices subject to change without notice.**

Circle 152 on Reader Service card.

Listing continued.

```
1000 REM  DETN. DAY
1010 HOME
1020 PRINT : PRINT "              DAY OF WEEK
1030 VTAB (5): PRINT "FOR WHAT CALENDAR DATE DO YOU WANT TO"
                                                       --- -- ----"
1040 VTAB (6): PRINT "DETERMINE THE DAY?  ENTER IN THE FORM"
1050 VTAB (7): PRINT "MM/DD/YYYY.  THE YEAR MUST BE AFTER 1582"
1060 INPUT Z$
1070 GOSUB 5000
1080 I = X - (7 * INT (X / 7))
1090 HOME
1100 VTAB (8): PRINT Z$;" IS A ";D$(I)
1110 VTAB (21): PRINT "PRESS 'RETURN' TO CONTINUE, 'E' TO EXIT"
1120 INPUT Z$
1130 IF Z$ = "E" THEN 10000
1140 GOTO 100
2000 REM  NUMBER OF DAYS BETWEEN DATES
2010 HOME
2020 PRINT : PRINT "          DAYS BETWEEN DATES"
2030 VTAB (3): PRINT "          ---- ------- -----"
2040 VTAB (5): PRINT "WHAT IS THE FIRST DATE?  ENTER IN THE"
2050 VTAB (6): PRINT "FORM MM/DD/YYYY"
2060 VTAB (8): PRINT "THE YEAR MUST BE AFTER 1582"
2070 INPUT Z$
2080 GOSUB 5000
2090 X1 = X:M1 = M:D1 = D:Y1 = Y:Z1$ = Z$
2100 VTAB (12): PRINT "WHAT IS THE SECOND DATE?"
2110 VTAB (13): PRINT "MM/DD/YYYY"
2120 INPUT Z$
2130 GOSUB 5000
2140 N =  ABS (X - X1)
2150 HOME
2160 VTAB (8): PRINT "THE NUMBER OF DAYS BETWEEN ";Z1$
2170 VTAB (10): PRINT "AND ";Z$;" IS "N
2180 VTAB (21): PRINT "PRESS 'RETURN' TO CONTINUE, 'E' TO EXIT"
2190 INPUT Z$
2200 IF Z$ = "E" THEN 10000
2210 GOTO 100
3000 REM  NUMBER OF THE DAY IN THE YEAR
3010 HOME
3020 PRINT : PRINT "        NUMBER OF DAY IN YEAR"
3030 VTAB (3): PRINT "        ------ -- --- -- ----"
3040 VTAB (5): PRINT "WHAT IS THE DATE?  ENTER IN THE FORM"
3050 VTAB (6): PRINT "MM/DD/YYYY.  THE YEAR MUST BE AFTER 1582"
3060 INPUT Z$
3070 GOSUB 5000
3080 X1 = X:M1 = M:D1 = D
3090 M = 1:D = 0
3100 GOSUB 6000
3110 N = X1 - X
3120 HOME
3130 VTAB (8): PRINT Z$;" IS DAY ";N
3140 VTAB (21): PRINT "PRESS 'RETURN' TO CONTINUE, 'E' TO EXIT"
3150 INPUT Z$
3160 IF Z$ = "E" THEN 10000
3170 GOTO 100
4000 REM  JULIAN DATE
4010 HOME
4020 PRINT : PRINT "              JULIAN DATE"
4030 VTAB (3): PRINT "              ------ ----"
4040 VTAB (5): PRINT "TYPE THE DATE IN THE FORM MM/DD/YYYY."
4050 VTAB (6): PRINT "THE YEAR MUST BE AFTER 1582"
4060 INPUT Z$
4070 GOSUB 5000
4080 X1 = X:M1 = M:D1 = D:Y1 = Y
4090 M = 1:D = 1:Y = 1583
4100 GOSUB 6000
4110 J = 2299238.5 + (X1 - X)
4120 HOME
4130 VTAB (8): PRINT Z$;" IS JULIAN DAY ";J
4140 VTAB (21): PRINT "PRESS 'RETURN' TO CONTINUE, 'E' TO EXIT"
4150 INPUT Z$
4160 IF Z$ = "E" THEN 10000
4170 GOTO 100
5000 REM  DATE SUBROUTINE
5010 FOR I = 1 TO  LEN (Z$)
5020 IF  MID$ (Z$,I,1) <  > "/" THEN  NEXT I
5030 REM  I IS THE POSITION OF THE FIRST'/'
5040 FOR N = I + 1 TO  LEN (Z$)
5050 IF  MID$ (Z$,N,1) <  > "/" THEN  NEXT N
5060 REM  N IS THE POSITION OF THE SECOND '/'
5070 M$ =  LEFT$ (Z$,I - 1):M =  VAL (M$)
5080 D$ =  MID$ (Z$,I + 1,N - I - 1):D =  VAL (D$)
```

Listing continued.

Listing continued.

```
5090 Y$ =  RIGHT$ (Z$,4):Y =  VAL (Y$)
5100  IF M >  = 1 AND M <  = 12 THEN 5140
5110  VTAB (15): PRINT "THE MONTH ";M;" DOESN'T EXIST."
5120  VTAB (17): PRINT "PRESS 'RETURN' AND START OVER."
5130  INPUT A$: GOTO 100
5140  IF D >  = 1 AND D <  = 31 THEN 5180
5150  VTAB (15): PRINT "A MONTH DOESN'T HAVE ";D;" DAYS."
5160  VTAB (17): PRINT "PRESS 'RETURN' AND START OVER."
5170  INPUT A$: GOTO 100
5180  IF Y >  = 1583 THEN 6000
5190  VTAB (15): PRINT "THE YEAR HAS TO BE AFTER 1582"
5200  VTAB (16): PRINT "YOU TYPED ";Y
5210  VTAB (17): PRINT "PRESS 'RETURN' AND START OVER."
5220  INPUT A$: GOTO 100
6000  REM   DETERMINE VALUE
6010  IF M > 2 THEN  GOTO 6040
6020 X = 365 * Y + D + 31 * (M - 1) +  INT ((Y - 1) / 4) -  INT (.75 *  INT (((Y - 1) / 100) + 1))
6030  RETURN
6040 X = 365 * Y + D + 31 * (M - 1) -  INT (.4 * M + 2.3) +  INT (Y / 4) -  INT (.75 * ( INT (Y / 100) + 1))
6050  RETURN
10000  VTAB (24): PRINT "END": END
```

```
                    TASKS AVAILABLE

               1--DAY OF THE WEEK
               2--DAYS BETWEEN DATES
               3--DAY OF THE YEAR
               4--JULIAN DAY
               5--EXIT PROGRAM


                       DAY OF WEEK
FOR WHAT CALENDAR DATE DO YOU WANT TO DETERMINE THE DAY?  ENTER
IN THE FORM MM/DD/YYYY.  THE YEAR MUST BE AFTER 1582.
7/4/1776
7/4/1776 IS A THURSDAY


                  DAYS BETWEEN DATES
WHAT IS THE FIRST DATE?  ENTER IN THE FORM MM/DD/YYYY
THE YEAR MUST BE AFTER 1582
7/24/1969
WHAT IS THE SECOND DATE?
3/15/1981
THE NUMBER OF DAYS BETWEEN 7/24/1961 AND 3/15/1981 IS 4252

                 NUMBER OF DAY IN YEAR
WHAT IS THE DATE?  ENTER IN THE FORM MM/DD/YYYY.  THE YEAR
MUST BE AFTER 1582
5/23/1981
5/23/1981 IS DAY 143

                      JULIAN DATE
TYPE THE DATE IN THE FORM MM/DD/YYYY.  THE YEAR MUST BE AFTER 1582
8/21/1981
8/21/1981 IS JULIAN DAY 2444837.5
```

Sample run. Examples of what the calendar program can do.

January 1, 4713 B.C. Because the Julian Day begins at noon, fractional values will be obtained. Thus, the Julian date for August 12, 1981 is 2 444 828.5.

The Program

To determine the number of days between dates, first assign a value for each date, and then find the difference of the values. The values are computed using the equations on lines 6020 and 6040. Y is the year, M is the month and D is the day. The day of the year is determined using the equation on line 1080. If I = 0, then the day is Saturday; if I = 1, Sunday, etc.

The subroutine beginning at line 5000 lets you enter the date in the form MM/DD/YYYY.

I wrote the program for an Apple II Plus, but you should be able to run it on other computers with only minor changes. ■

DOS Depression

A simple method of avoiding a particularly frustrating DOS error is our first offering for inCider's Hints 'n' Techniques department.

by David W. Dilks

While working on a several-hundred-line program on my Apple II, I began receiving inexplicable DOS errors: FILE NOT FOUND when opening a new file for output and APPEND commands writing over existing data at the beginning of files.

To locate the cause of this annoying problem, I began deleting one program line at a time until all that remained was the program shown in the program listing.

```
10  D$ = CHR$(4)
20  NA$ = "DATA"
30  GET A$
40  PRINT D$; "OPEN"; NA$
50  PRINT D$; "WRITE"; NA$
60  PRINT D$; "CLOSE"; NA$
70  END
```

Program listing.

Try typing in and running this program yourself. If the file named Data doesn't exist on the disk, this program will terminate with a FILE NOT FOUND error. However, when line 30 is deleted, the program runs smoothly. Another way to solve the problem is to insert a dummy Print statement between lines 30 and 40.

Moral: To avoid Apple DOS problems, locate at least one Print statement between each Get command and any disk operation. ■

Address correspondence to David W. Dilks, Project Engineer, LTI, Limno-Tech Inc., 15 Research Drive, Ann Arbor, MI 48103.

Coming Next Month

Shape tables. . . vector graphics. . . do these defy comprehension? Does the mere mention of such send you into fits of deep depression? Fear no longer. The Fantastic Fudge, of Hi-Res Secrets fame, initiates his graphics column next month, and gifts all with a shape creator that'll have you wondering why you ever thought Apple graphics was confusing.

Also reviewed in the February *inCider* is a monster of a word processor, a sparkling speller, some game grabbers, and quiescent drives. EPROM programming, a Pascal Plunge, and the assorted craziness of Paul Raymer make February an issue you just cannot do without! Not only do two machines converse with each other, but Wayne will have another chat with you, and screen presentation of text is simplified.

And just think. . . if you'd subscribed, you wouldn't have to walk to your local magazine dealer to purchase the February *inCider*. Besides that,. . . your dealer may have already SOLD OUT!

The Applesoft Adviser

This and subsequent articles will provide a tutorial review of Basic. I plan to present material and examples that will suit novice programmers as well as some subroutines and techniques more sophisticated users will appreciate.

by Dan Bishop

So you've bought yourself an Apple. Now you're wondering if a microcomputer can do more than play games and run marginally useful business programs that must do until what you really need comes along. And play games.

Congratulations on buying a microcomputer! And don't despair over the dearth of suitable software. Learn to write your own programs. It is easier than you think and you might find out that your business software meets your needs after all. A little Basic may be all you need to modify those programs so they operate exactly to your specifications.

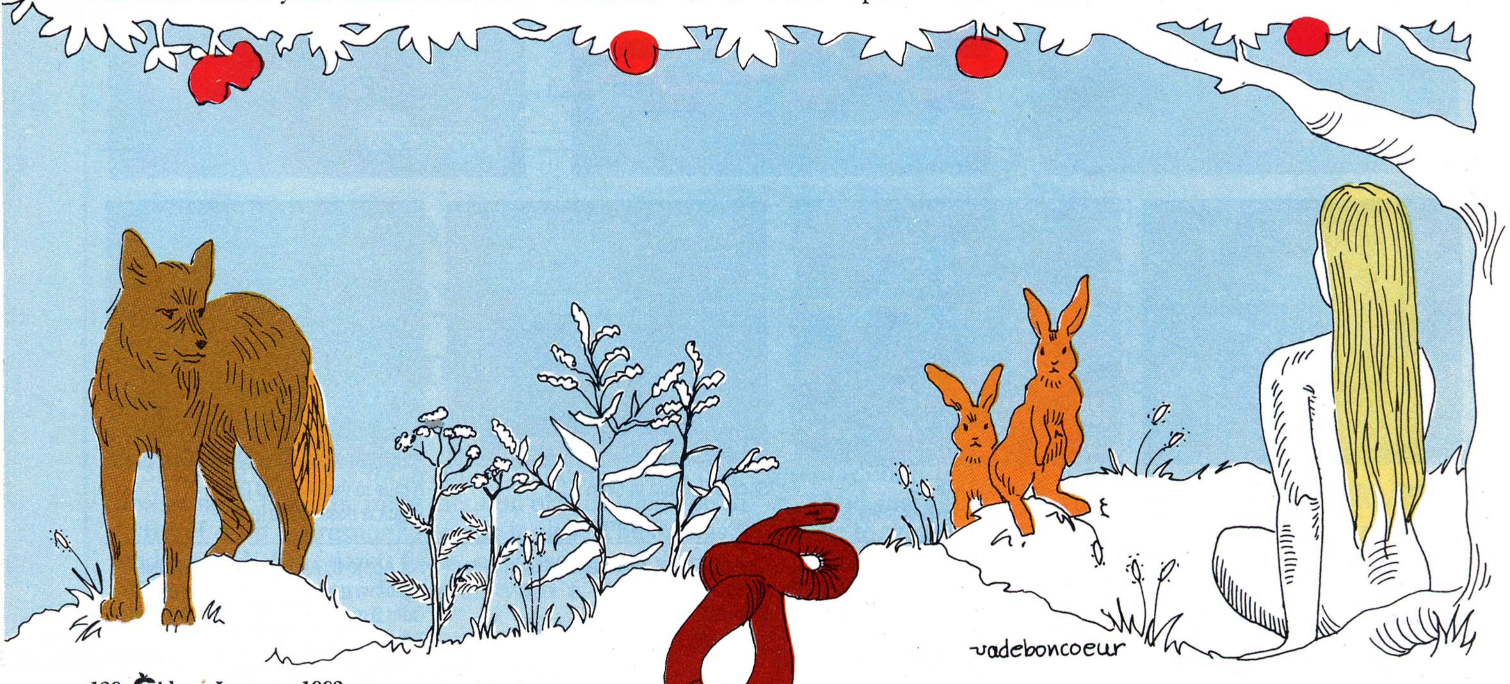
Just What Are Variables?

When you meet someone new, you instinctively listen closely to the person's name, and make a point to remember the name, particularly if you anticipate future meetings. The name serves as a "tag" to identify that person. Should you spot him in a crowd and wish to attract his attention, you could call his name and hope he could hear you over the din.

Now suppose several years pass, and you recognize him walking across the street. He has grown older, snowy-bearded and balding. He is definitely not the same person you remembered, yet when you call his name, he turns, crosses the street, and joins you for coffee at a nearby cafe.

In a program, a variable is a name

Address correspondence to Dan Bishop, Custom Comp, PO Box 429, Buena Vista, CO 81211.



assigned to some quantity you wish to manipulate within the program. Not all such quantities need a variable name. For example, if the number 2.5 is used within a program only once, and its value never will change, then to assign it a name is unnecessary. Simply use the numeric digits, 2.5.

On the other hand, suppose you enter the name "Walton's Widgets Inc." frequently throughout a program. Here is a likely candidate for a variable name, say WW$. (More about selecting names below.)

Once you have assigned the value "Walton's Widgets Inc." to the tag WW$, then whenever you use WW$ anywhere in the program, the computer will come up with "Walton's Widgets Inc." This saves time writing the program, and memory space as well. Another advantage comes when you learn that Walton just sold his Widget factory to Wayne Green and the company name becomes "Green's Gidgets Inc." If we had not used WW$ to represent the company name, the name might need to be changed in two dozen places throughout the program. By using a variable name, you need only change what WW$ stands for.

Most programs involve calculations that use numeric values that may change from one time to the next, or sometimes even change several times within the program itself. It would be impossible to write a program that allowed this sort of thing without using variable names. Suppose, for example, you are counting the number of paychecks printed in a payroll program. Let PC% be the "tag," or variable name, representing the number of paychecks. Before starting the printing process, the program will set the value of PC% to zero; then, as each paycheck is printed, it adds 1 to the previous value of PC%. In this way PC% is kept current, and, at the end of the program, it will contain the number that represents the total number of paychecks printed. A variable used in this type of application is called an accumulator or counter, and the instruction that accomplishes the incrementing function is simply:

```
PC% = PC% + 1
```

This formula reads: "To derive the new value for PC%, add 1 to the old value for PC%." This type of formula makes no sense in algebra (how can anything be equal to itself plus 1?), but in Basic the equal sign in an equation like this means "assign the result of the following calculation to the specified variable." Such equations are called assignment statements. The value for PC% in this example changes after each new paycheck is printed, but the computer doesn't care.

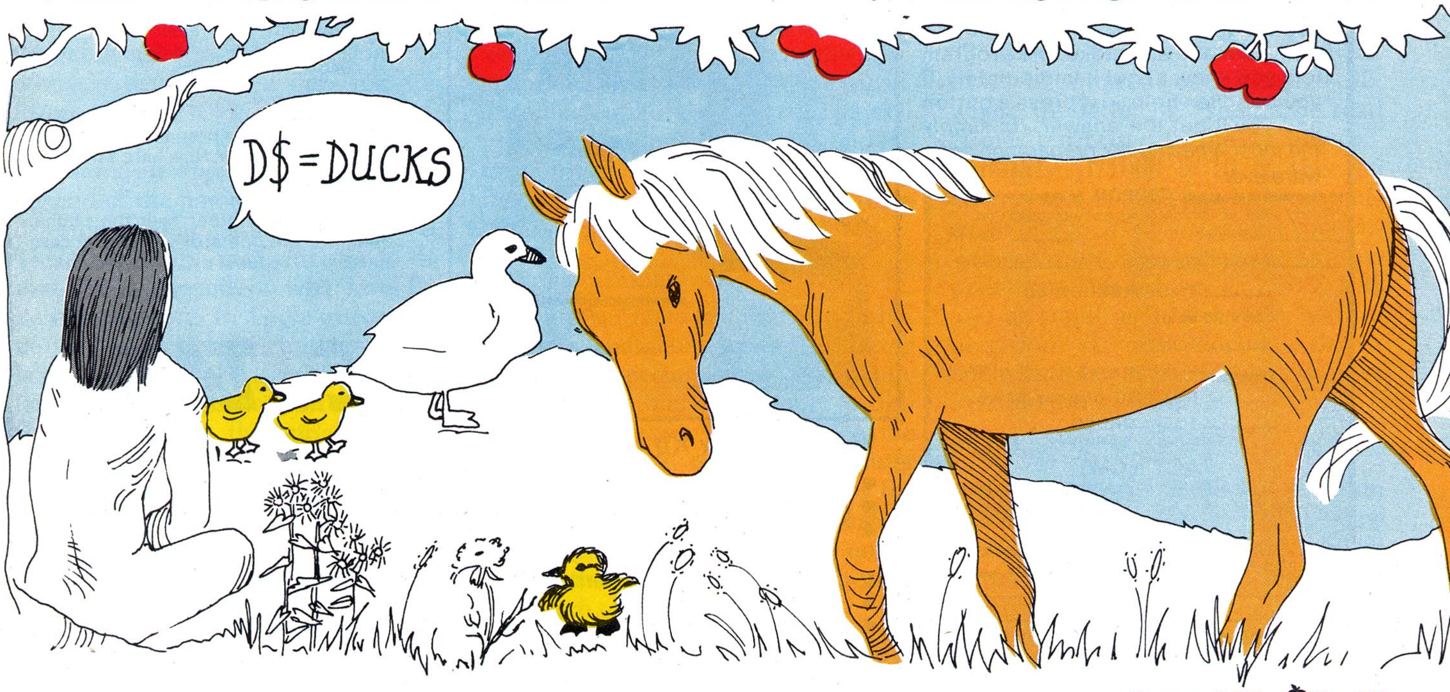
Whenever PC% is called for in the program, the computer pulls out the current value of PC% and uses it. Just as your aging friend still responded to his name though he had changed through the years, the variable PC% still represents the paycheck count, even though its value changes as the program runs.

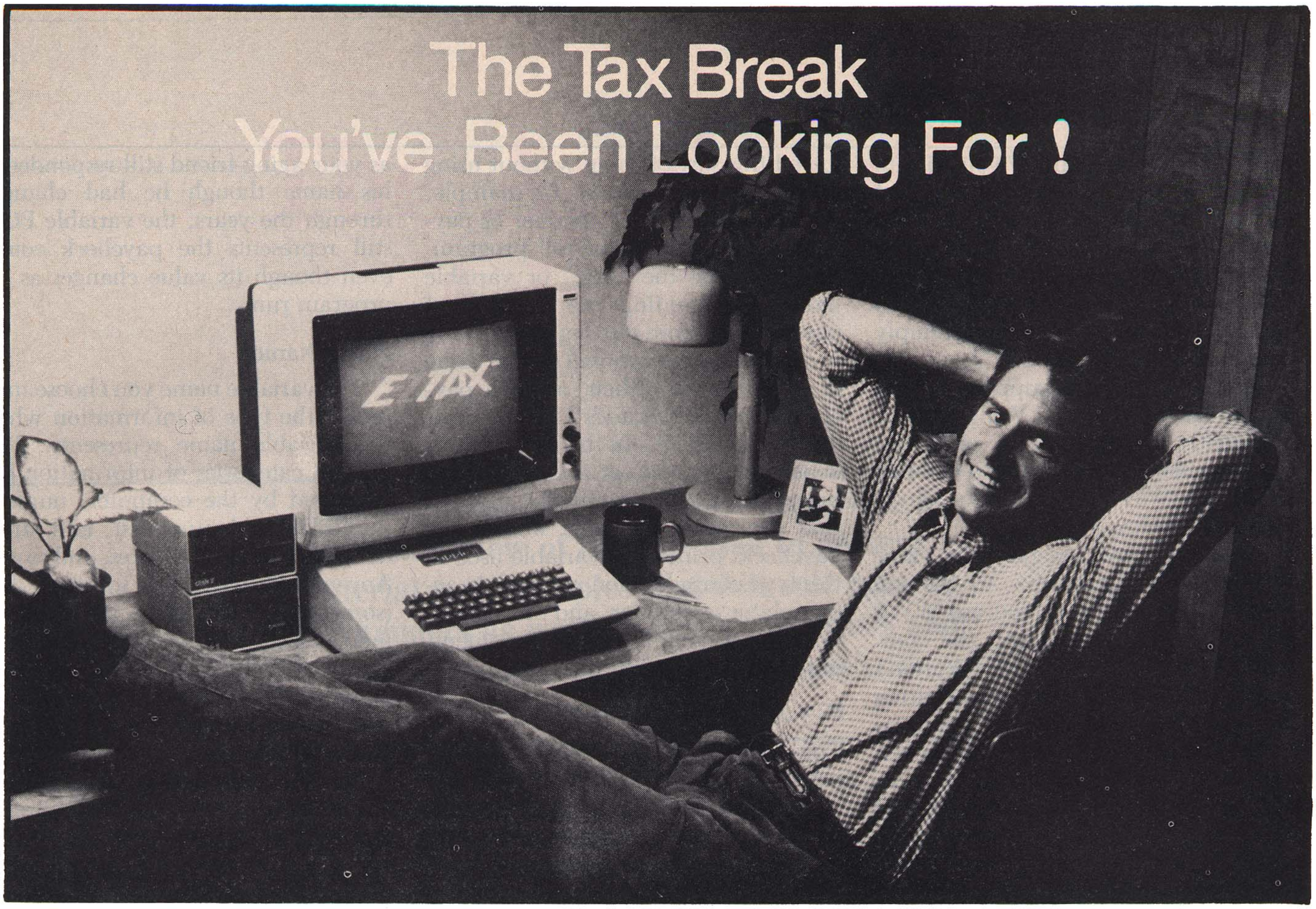
Pick a Name

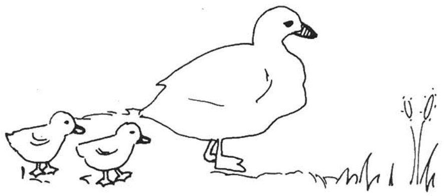
The variable name you choose must reflect the type of information which the variable name represents. Two general categories of information can be stored by the computer: numeric (just plain old numbers), and string data (such as names, dates, and so on). Any variable name used for string data (sometimes referred to as alphanumeric data or literal data) *must* end with the dollar sign. In the example above we chose WW$ to represent the name of the company. The dollar sign was *not* optional.

Numeric information consists strictly of numbers, with an optional + or − sign and a decimal point. The Apple allows you to represent numeric information in two different forms. If the value(s) represented by a given variable name will always be whole numbers (integers) with no decimal fractions, then the variable name should end with a percent sign, such as PC%.

The example above matches the requirements for an integer variable, and consequently applies the percent sign as part of the variable name. Since denoting integer variables with the




The Tax Break
You've Been Looking For !
E-Z TAX



percent sign is optional, many programmers drop the notation. However, clearly specifying to the computer just which variables are for integer values saves memory space, and makes the program run faster and more efficiently. You should note that Applesoft does *not* allow the use of integer variables as counters in For. . . Next loops or in DEF statements.

Numeric variable names that represent numbers that are not necessarily integers (real numbers) are given without any symbol following the variable name. Thus the value 5.62135 may be assigned to a variable name such as RZ. Real numbers require more memory space, so use real numeric variables sparingly. Some Basics allow two types of real numeric variables: single precision (denoted by an exclamation point following the variable name, such as RZ!); and double precision (denoted by a # following the variable name, such as RZ#). Applesoft II allows up to ten-digit precision for real numbers, so the lack of a double precision mode is probably insignificant unless you are dealing with the national debt!

(Another feature of some Basics is a DEF function which allows the programmer to specify variable types *en masse* at the beginning of the program. For example, DEFINT L–Q would demand that all variables beginning with the alphabetic characters L, M, N, O, P or Q are integers (unless otherwise indicated by a $, ! or #). The Apple user converting a program from another system must watch for this feature.)

We have seen that variable names may or may not have type identifiers at the end of the name. Now we need to determine what names are allowed. This varies from one computer to the next. It also depends on what level Basic you are using. In Apple Integer Basic you may use any combination of letters and numbers of reasonable length. The only restriction is that the first character in the variable name must be alphabetic.

On the other hand, Applesoft Basic allows you the same latitude as Integer Basic concerning variable name length, but recognizes only the first two characters of your name and ignores the rest. Thus PAY, PART and PAY1982 will all be considered to be the same variable, PA. As with Integer Basic, the first character must be alphabetic. I prefer only one or two characters for variable names within a program. Although not quite as descriptive, it saves memory, and avoids my inadvertently using the same first two characters for two different variables.

One other important restriction on the variable name you choose is that you must avoid using a Basic keyword. The computer would confuse your variable name with a Basic command. These forbidden words are called "reserved words," and a list of them can be found in the Applesoft manual on page 122. Using a reserved word as a variable name will produce a syntax error when you attempt to run the program.

In fact, using a reserved word as any *part* of your variable name will cause problems. For example, the computer will read the IF in SPIF as a command keyword. You can avoid this problem by restricting your variable names to two characters. The few two-character keywords (IF, OR, FN, ON, AT, GR and TO) are easy to remember. Errors caused by using a keyword as part of a variable name are difficult to diagnose. It is seldom apparent when inspecting the problem line just what the problem really is. All common syntax problems can be tested, yet the program insists you have a syntax error in that line.

Variable names with different type identifiers attached to them may appear in the same program without risk of confusion. For example, our payroll program could use PC%, PC$ and PC to represent three different items and the computer would track all three separately without problems.

Variable Limitations

String variables may be any alphanumeric character string, including most of the special characters. Usually the quotation mark is the only forbidden character for strings, since the quotation mark tells the computer where the beginning and end of the string of information are located. However, a string variable may not be longer than 255 characters. It may have zero characters as well.

For example, ZZ$ = " ", where the two sets of quotes have no space between them, is a valid assignment statement, making the variable ZZ$ represent a null string. When assigning a value to a string variable, enclose the string of characters in quotation marks. Thus the statement NM$ = GEORGE WASHINGTON would produce a type mismatch error, while NM$ = "GEORGE WASHINGTON" would be acceptable.

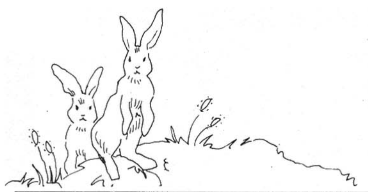
Integer variables are limited to representing values between −32767 and +32767. For those who may be interested in the reason for integer variables being limited by such odd numbers, you will note that 2 raised to the 15th power is 32768. Integers are represented by two bytes in the microcomputer memory, which correspond to 16 binary digits (bits). If one of these bits is used to represent the sign of the number, only 15 are left to represent the number itself, thus the limitation of ±32767.

Real variables may have up to nine decimal digits. You are not limited to 999 billion, however. It merely indicates the degree of precision that such a number can have. The tenth digit will always be rounded and will not be displayed. For numbers greater than 999 billion, and smaller than 0.01, the computer uses exponential (or scientific) notation, expressing the number as some number times the appropriate power of 10.

For example, 10 squared is 100, so 893.22 can be represented as 8.9322 times 10 squared, or 8.9322 E+02, where the letter "E" should be read "times ten raised to the . . . power." Similarly, 1/1000 is 10 to the −3 power, so 0.003958 can be represented as 3.958 E−03. Using exponential notation, the computer can express numbers as large as 1.00 E+38 and as small as 1.00 E−38, with a precision of nine decimal digits. Such limitations should satisfy most earthbound humans!

Clearing Variables

As you might guess, a program that's been running for some time probably has a whole list of variables stored in its memory. You might reasonably ask whether the variables stored in one program could get in the way of the variables needed by a later



program.

The answer is that when you type the Run command to execute a program, all variables are reset to zero. Thus there is no chance for carry-over from one program to the next. Another Basic command, Clear, can be used within a program to accomplish the same task.

Variables can also be used in immediate (or command) mode. Try this with your computer on and ready. Type in A = 25 and press return. Now type in PRINT A, SQR(A) and press return. You should see the numbers 25 and 5 (the square root of 25) appear on the screen. Any time you are using a computer as a calculator, you will find it very useful to store certain numbers as variables for later (or repeated) recall.

String vs Numeric

Numbers may be represented with either of the numeric variable types or with string variables. An important difference exists between the two representations. If a number is represented by a string variable name, it is taken as a literal—the computer will not recognize it as a number, and it will be interpreted as merely a string of characters. Such a representation may be suitable for zip codes, for instance. If you want to carry out any type of mathematical operation on the numbers, however, use a numeric variable name to represent the number. For example, the program line:

```
A$ = "1.355":B$ = "2.8661": PRINT A$ + B$
```

will print 1.3552.8661, since the numbers are represented by string variables and the + sign, when used with string variables, merely concatenates the two strings.

Lining Up Decimal Points

It is possible to convert numeric data to string information and back again, and on occasion this is useful.

Some Basics have an output formatting instruction such as a Print Using instruction in which the programmer can precisely specify the appearance of the printed output, allowing for numbers like 1.33, 2.8, 1000 and 0.01 to be printed in a column with all of the decimal points correctly aligned.

Apple Basic does not have this capability, so you must convert all of the numbers to strings (using the STR$( ) function), then adjust the strings so that each literal numeric expression has the same number of characters to the left of the decimal point. You can best handle this using a defined function or a subroutine within the program. If you choose a subroutine, the same subroutine can also round all numbers to the desired precision before converting them to strings and carrying out the string manipulations.

Suppose that X is the variable representing a dollar amount you're working with.

```
10 D% = 2
11 X = INT(X*10∧D% + .5)/INT
   (10∧D% + .5)
12 ZZ$ = STR$(X)
13 Z$ = "."
14 Z% = 0
15 FOR I = 1 TO LEN(ZZ$)
16       IF MID$(ZZ$,I,1) = Z$ THEN Z% = I
17 NEXT I
18 IF Z%<>0 THEN 21
19       ZZ$ = ZZ$ +
   LEFT$(".0000000",D% + 1)
20       GOTO 24
21 IF LEN(ZZ$) - Z% = D% THEN 24
22       ZZ$ = ZZ$ + "0"
23       GOTO 21
24 ZZ$ = RIGHT$("          " + ZZ$, 10)
25 RETURN
```

The value stored in X has now been converted to a string variable ZZ$ of exactly ten characters (including padding spaces to the left), with exactly two digits to the right of the decimal point.

This short subroutine may be used in any program containing financial information. D% is assigned the value of 2, to represent the number of places to the right of the decimal point; two places should be reserved if the numbers are to be expressed in dollars and cents. The number is then rounded to the nearest cent, converted to a string using the STR$( ) function, and then, in line 16, the position of the decimal point is determined. This is necessary because a number such as 35.00 would be represented as simply 35, while 27.60 would be represented as 27.6.

To produce numbers that always have two digits (including trailing zeros) to the right of the decimal point, you must determine if there is a decimal point and, if so, just where it is. The value of Z% will be 0 if there is no decimal point; otherwise it will tell us how many digits there are to the left of the decimal point, plus 1.

This value is compared to the full length of ZZ$ using the LEN( ) function, and if it is found that there are fewer than D% digits to the right of the decimal point, a literal 0 is attached to the end of ZZ$ and the new ZZ$ is similarly evaluated. Finally, in line 24, ZZ$ is attached to the end of a string of ten blank spaces, and the rightmost ten characters are assigned anew to ZZ$. This effectively pads your literal number with as many blank spaces in front of the first digit as is necessary.

In order to use this subroutine on a number represented as, say, PR, simply assign PR to X, (X = PR), and Gosub 10:

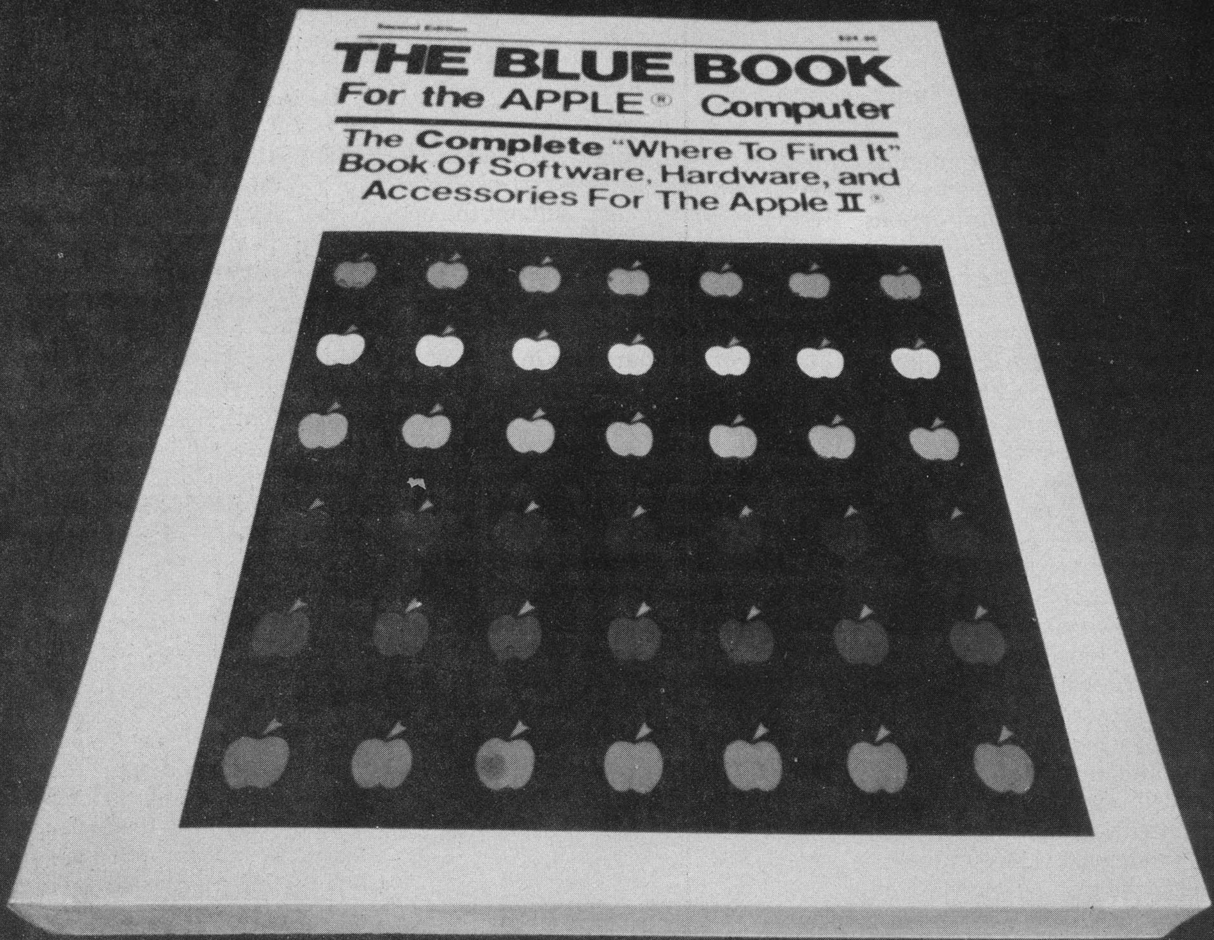
```
X = PR
GOSUB 10
PRINT ZZ$
```

You may wish to change the value assigned to D% in line 10 to allow for more than two places to the right of the decimal point. You may also change the number of blank spaces and the number of characters taken by the RIGHT$ function in line 24, particularly if you are dealing with different applications. In any case, the displayed results will look more professional with each number properly lined up in columnar fashion.

Strings can be converted back to their numeric values using the VAL( ) function. For example, if X$ = "3.14159", then X = VAL(X$) will assign the numeric value 3.14159 to the variable name X, so the value can be used in mathematical operations. This situation most commonly occurs as a result of an Input statement that requires the operator to type in a response to a displayed question. If the variable used with the Input statement is a string variable, (e.g., INPUT X$), but the numeric information typed in must be evaluated as numeric data, then the VAL( ) function comes to the rescue.

Another useful application of the

Your best Apple® accessory



The Blue Book™ for the Apple® Computer is **the accessory** that can save you time and money searching for other Apple® accessories. The Blue Book™ is a complete, "where to find it" directory of software, hardware, and accessories with over 2350 product listings in 57 different categories. The Blue Book™ helps you choose new programs for your Apple II® by careful comparison shopping. This is the book that puts you directly in touch with more than 450 software and hardware manufacturers. Whether you are a hobbyist, a businessman, or an educator, The Blue Book™ is a "must-have" reference book, and a perfect companion for your Apple.®

Apple and Apple II are trademarks of Apple Computer Co.
The Blue Book is a trademark of WIDL Video, Chicago.

Please ship me_____copies of **The Blue Book**™ for the Apple® Computer at $24.95 each plus $2.00 shipping and handling charge.

- ☐ Check or money order enclosed
- ☐ Visa
- ☐ MasterCard

Please print clearly

Please make check payable to and mail to:
WIDL Video
5245 W. Diversey
Chicago, IL 60639
(312) 622-9606

card account # exp. date

signature

name

address

city state zip

( )
telephone

Circle 151 on Reader Service card.

```
PRESS:   <A> TO ADD ACCOUNTS      <D> TO DELETE ACCOUNTS
         <E> TO EDIT ACCOUNTS     <V> TO VIEW ACCOUNTS
         <P> TO PRINT ACCOUNTS    <M> TO RETURN TO MASTER MENU
         <?> FOR HELP             <X> TO END PROGRAM
-------------------------------------------------------------
                          CHOICE...
```

Figure 1. Menu display using mnemonic option selection keys.

VAL( ) function is in determining the month when a date has been entered in 8-byte form as MM/DD/YY. For example, if DT$ = "06/12/82", then DT = VAL(DT$) would assign the numeric value +06 to DT.

Menu Selection Routine Made Easy

Many applications programs are designed to be run by noncomputerists. Such programs are usually menu driven, which means that a list of options corresponding to the different program functions is displayed on the screen. The operator selects one option and presses the appropriate key. The program then branches to the routines that deal with the selected function.

The branching instruction must be numeric, so many menus display the options with numbers, and the operator must press a number key to select a given option. There are two drawbacks to this approach. First, if there are more than ten options, the operator must press two number keys and the return key, requiring three keystrokes. The second difficulty lies in the fact that it is easy to accidentally hit the wrong key when all of the option choices are numeric.

One way to avoid these problems and make program operation smoother is to use mnemonic letters for the responses. Thus the menu display in Figure 1 instructs the operator to press one of eight keys. The following subroutine will accept only the desired eight responses, ignoring any other keystroke. Furthermore, by using related alphabetic keys, the chance for mistake is reduced. You can respond to any of more than ten choices with a single keystroke.

The subroutine below will handle this situation well. But before using it, the following lines must be placed immediately after the series of print statements which set up the menu display in the first place.

```
PRINT"CHOICE...";
ZZ$ = "ADEVPM?X"
GOSUB 30
```

ZZ$ is a list of the acceptable responses. The subroutine which handles the keyboard response is:

```
30 GET Z$
31 Z% = 0
32 FOR I = 1 TO LEN(ZZ$)
33      IF MID$(ZZ$,I,1) = Z$ THEN Z% = I
34 NEXT I
35 IF Z% = 0 THEN 30
36 PRINT"**   ";Z$;"   **"
37 RETURN
```

The Get instruction acts as an Input statement for a single keystroke, and responds immediately without the operator having to press return. Once a key has been pressed, the program proceeds to the next instruction. In this routine, a match is sought between Z$ and one of the characters in ZZ$. If no match is found, the program returns to the Get statement (line 35). If a match *is* found, then the selection is printed on the screen and the program returns to the instruction that follows the Gosub 30 instruction.

The return is made with a numeric value, Z%, that corresponds to the selection in the order in which it appeared in ZZ$. Thus Z% = 1 if the selection had been A, Z% = 2 if the selection had been D, etc. On returning to the main program sequence, you can use a statement such as:

```
ON Z% GOTO...., ...., ...., ...., ....,
...., ...., ....
```

where each of the ellipses corresponds to the program line number to which the program should branch, based on the value of Z%.

Incidentally, the loop in lines 32–34 can be replaced in some Basics by the INSTR( ) function. If you are converting such a program to Apple Basic, the INSTR( ) instruction equivalent to lines 32–34 above would be Z% = INSTR(ZZ$,Z$), with no loop necessary.

A careful reader might have noticed a similarity between the two subroutines presented above. Lines 14–17 in the first example are identical to lines 31–34 in the second example. I did this to show that two seemingly different functions within a program might share common portions. Program length can be considerably reduced if the programmer takes advantage of these similarities, incorporating them into a single subroutine and trying, where possible, to use common variable names. If you use the following lines:

```
40 Z% = 0:FORI = 1TOLEN(ZZ$):IFMID$
   (ZZ$,I,1) = Z$THENZ% = I
41 NEXTI:RETURN
```

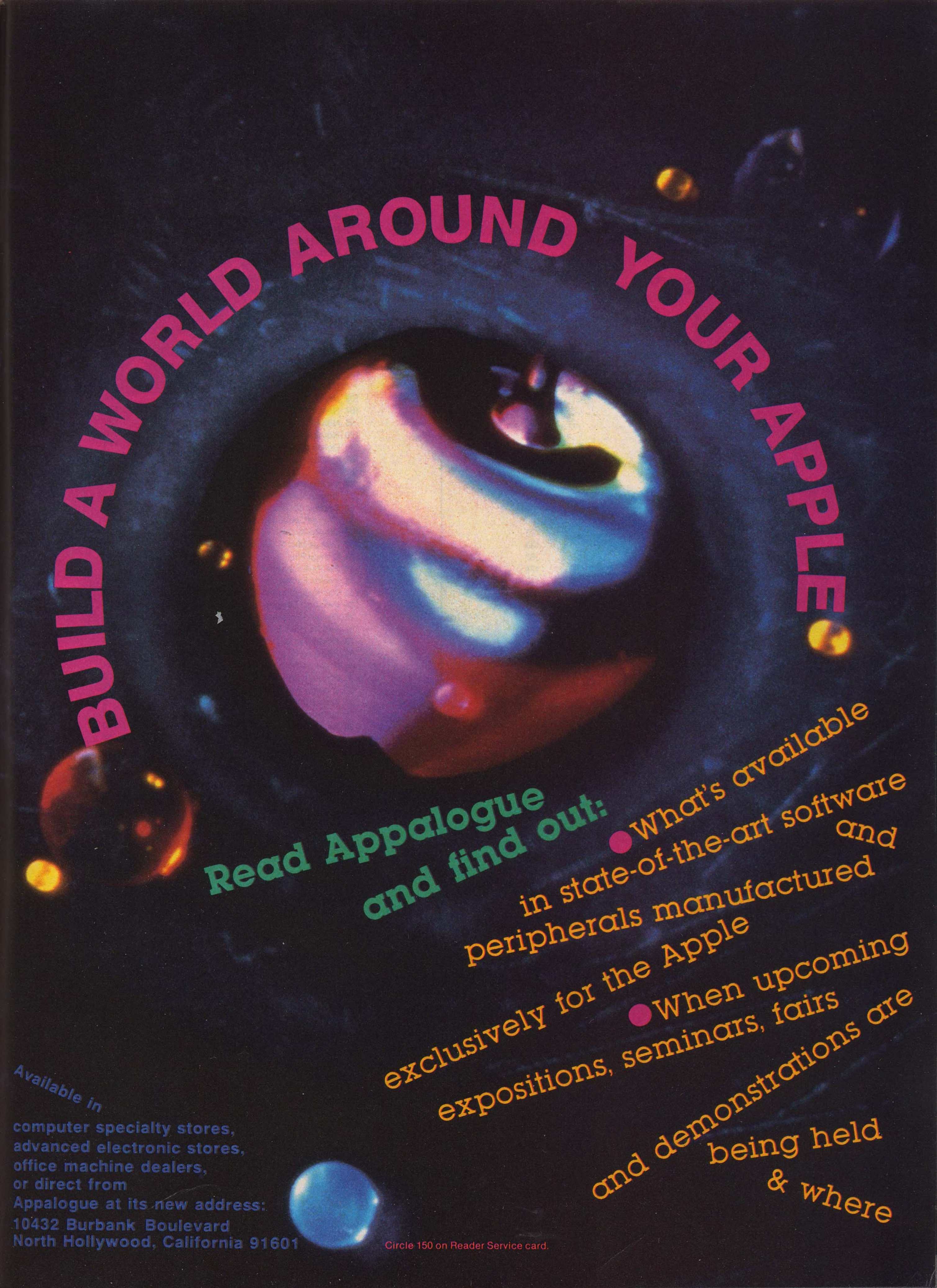
then change line 14 in the first example to Gosub 40, deleting lines 15–17; and change line 31 in the second example to Gosub 40, deleting lines 32–34. You'll have reduced the program's length by about 50 bytes.

A good part of the savings (20 bytes) came from using a multiple statement line (line 40) in which several statements share the same line number, each separated from the others by a colon. Some Basics do not allow this; others use a different character for a separator (such as a back-slash). Although such a line is more difficult to read, it does increase program efficiency. Remember, each line in your program, regardless of the actual line number used, requires five bytes of overhead.

Conclusion

In this article I've discussed how variables are named and used, and the limitations in naming variables and in assigning values to them. Several string functions appear in the examples without detailed explanation, showing some of the more advanced operations you might attempt. Of course the only real way to understand these concepts is to play around with them yourself. Obviously, if you have read through this article, you must be interested in using Basic as a tool. The next step (practice, practice, practice) is up to you.

Another side to variable names involves the use of arrays. Arrays allow simplification of many common program operations, and are frequently useful. Any type of variable, (string, integer or real) may be used as an array, and the same rules presented above for variable names apply to the names chosen for array variables. The only difference is that array variables are dimensioned and are followed by a subscript (in parentheses). My next article will be devoted primarily to the use of arrays in program applications. ■


BUILD A WORLD AROUND YOUR APPLE
Read Appalogue
and find out:
• What's available
in state-of-the-art software
and
peripherals manufactured
exclusively for the Apple
• When upcoming
expositions, seminars, fairs
and demonstrations are
being held
& where
Available in
computer specialty stores,
advanced electronic stores,
office machine dealers,
or direct from
Appalogue at its new address:
10432 Burbank Boulevard
North Hollywood, California 91601
Circle 150 on Reader Service card.

Index to Advertisers

inCider™ Green's Apple* Magazine

*This advertiser prefers to be contacted directly.

For further information from our advertisers, please use the Reader Service card.

INTRODUCING

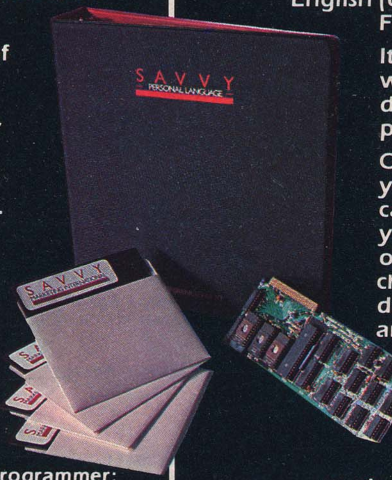
SAVVY™



. . . the PERSONAL LANGUAGE™ system that mirrors your commands using your own words!

What SAVVY is —

- SAVVY is a miraculous new information handling system.
- SAVVY is an automatic database management system.
- SAVVY is a new level of machine intelligence.
- SAVVY, part hardware, part software, is the beginning of truly "Personal Computing".
- SAVVY comes with: General Ledger, Accounts Receivable, Accounts Payable, Payroll, Mailing List, Document Writer and Inventory Control.



Trademark: SAVVY, Robot Programmer: Excalibur Technologies Corporation.

Personal Language: SAVVY Marketing International.

What SAVVY does —

Through SAVVY, you and your computer talk to each other in your own natural, conversational English (or Spanish, or French, etc.).

It learns from you what you want done in your own personal language.

Once SAVVY learns your language it can create any file you wish. Input, output, additions, changes and deletions are arranged for you.

SAVVY's "Robot Programmer" has been trained to write 100% of the programs needed to manage your database information.

SAVVY runs CP/M™ and Apple DOS.

What YOU discover —

You'll discover that SAVVY recognizes your personal words, even if misspelled, or even if you use a phrase never used before!

SAVVY continues to grow through use to become better and better at understanding your commands.

Eventually, you will see SAVVY as a mirror to your own way of thinking and working. It is a re-definition of "user-friendly".

SAVVY, it's the first system that truly means "personal computing".

SAVVY is like no other system on earth.

SAVVY cost $950.

Seeing is believing. SAVVY is on display at selected computer retail locations. Call for the name of your nearest dealer.

CP/M is a trademark of Digital Research Corp.

Apple is a Trademark of Apple Computer Inc.

Circle 7 on Reader Service card.

100 South Ellsworth Street, 9th Floor, San Mateo, CA 94401 (415) 340-0335

SAVVY

MARKETING INTERNATIONAL

Piano Plinkin'

Did you think the Apple III was just for routine office chores? Not so! With this short program you'll soon be in the music business.

by Billy Moore



Piano (see Program listing) turns your Apple III keyboard into a one-octave piano-type keyboard complete with properly placed "black keys." You can play melodies in real time while the notes are displayed on the monitor. And once you've entered a melody, you can recall it by pressing P or save it on disk by pressing N—so you can insert tunes easily into other programs.

The menu also includes simple edit features and variable speed and volume. Before trying this program, make sure an ".Audio" driver is included in the driver file. Also, the program as written accepts only uppercase letters to represent notes. Although I wrote the program in Apple III Business Basic, you can modify it easily to run on other computers with programmable sound. ■

Billy Moore (1889 Chantilly Lane, Hayward, CA 94541) is an eighth grade student at Head-Royce School in Oakland.

Program listing. Piano keyboard program in Apple III Business Basic.

```
10    PRINT CHR$(1);:HOME
20    GOSUB 450
30    REM           INITIALIZE
40    OPEN#1,".AUDIO"
50    DIM K$(12),P$(12),N(12),R(99):R=0:V=60:M=0:L=20
60    FOR J=0 TO 12:READ K$(J):NEXT J
70    FOR J=0 TO 12:READ N(J):NEXT J
80    FOR J=0 TO 12:READ P$(J):NEXT J
90    REM           INPUT
100    GET A$
110    REM           CHECK FOR ANY SPECIAL KEYS
120    IF A$="1" AND V>0 THEN V=V-1:GOTO 100
130    IF A$="2" AND V<60 THEN V=V+1:GOTO 100
140    IF A$="B" AND R>0 THEN R=R-1:PRINT"X ";:GOTO 100
150    IF A$="3" THEN V=60:GOTO 100
160    IF A$=" " THEN PRINT"/ ";:GOTO 265
170    IF A$="P" THEN GOTO 360
180    IF A$="C" THEN R=0:GOTO 100
190    IF A$="X" THEN HOME:GOTO 100
200    IF A$="N" THEN GOTO 720
210    IF A$="M" THEN GOTO 840
215    IF A$="Z" THEN PRINT:INPUT"SPEED (1=FAST, 20=NORMAL, 300= SLOW) ? ";L:GO
       TO 1000
220    IF A$="Q" THEN PRINT CHR$(1):HOME:CLOSE#1:END
230    FOR J=0 TO 12:IF A$=K$(J) THEN 280
240      NEXT J
250    IF A$="1" OR A$="2" OR A$="3" THEN GOTO 100
260    PRINT"/ ";
265    R(R)=0:R=R+1:IF R=299 THEN PRINT CHR$(7);:R=0:GOTO 100
270    GOTO 100
280    R(R)=N(J):R=R+1:IF R=299 THEN PRINT CHR$(7);:R=0:GOTO 100
290    REM           PLAY NOTE N(J)
300    PRINT#1;CHR$(128);CHR$(V);
310    PRINT#1;CHR$(N(J)-256*INT(N(J)/256));CHR$(INT(N(J)/256));
320    PRINT#1;CHR$(L-256*INT(L/256));CHR$(INT(L/256));
330    PRINT A$;" (";P$(J);") ";
340    GOTO 100
350    REM           PLAY NOTES IN MEMORY ( R(X) )
360    VOL=V
370    FOR X=0 TO R-1
380      IF R(X)=0 THEN VOL=0
390      PRINT#1;CHR$(128);CHR$(VOL);
400      PRINT#1;CHR$(R(X)-256*INT(R(X)/256));CHR$(INT(R(X)/256));
410      PRINT#1;CHR$(L-256*INT(L/256));CHR$(INT(L/256));
411      IF M=0 THEN GOTO 420
412      FOR J=0 TO 12:IF R(X)=N(J) THEN GOTO 415
413        NEXT J
414      IF R(X)=0 THEN PRINT"/ ";:GOTO 420
415      PRINT K$(J);"( ";P$(J);") ";
420      VOL=V
430      NEXT X
435    M=0
440    GOTO 100
450    REM           INSTRUCTIONS
```

Listing continued.

Listing continued.

```
460   PRINT USING"80c";"PIANO"
470   PRINT TAB(21);"     W      E       T      Y      U"
480   PRINT TAB(21);" A      S      D  F     G     H    J  K"
490   PRINT
500   PRINT TAB(21);"     C#     D#        F#     G#     A#"
510   PRINT TAB(21);" C      D      E  F     G     A     B  C"
520   PRINT:PRINT USING"80c";"COMMANDS"
530   PRINT TAB(10);"P = Play song in memory",
540   PRINT"B = Erase last note played"
550   PRINT TAB(10);"C = Clear memory         ",
560   PRINT"X = Clear screen below"
570   PRINT TAB(10);"1 = Lowers volume        ",
580   PRINT"M = Load song from disk"
590   PRINT TAB(10);"2 = Raises volume        ",
600   PRINT"N = Save song on disk"
610   PRINT TAB(10);"3 = Set volume to max. ",
615   PRINT"Z = Speed (1 - 300)"
620   PRINT TAB(10);"[space bar] = Rest       ",
630   PRINT"Q = Exit program"
640   PRINT CHR$(2);
650   RETURN
660   END
670   REM          BUTTONS, NOTE VALUES,& NOTES
680   DATA A,W,S,E,D,F,T,G,Y,H,U,J,K
690   DATA 1946,1836,1733,1636,1544,1457,1376,1298,1226,1157,1092,1031,973
700   DATA "C","C#","D","D#","E","F","F#","G","G#","A","A#","B","C"
710   END
720   REM          SAVE SONG ON DISK
730   HOME:INPUT"WHAT NAME? ";N$:IF LEN(N$)>15 THEN GOTO 730
735   IF N$="PIANO" THEN PRINT CHR$(7):GOTO 730
740   PREFIX$=".D1/"
750   CREATE N$, DATA
760   PREFIX$=".D1/":OPEN#2 AS OUTPUT,N$
765   WRITE#2;R
770   FOR T=0 TO R-1
780     WRITE#2;R(T)
790     NEXT T
800   CLOSE#2
810   PREFIX$=".D1/":LOCK N$
820   HOME
830   GOTO 100
840   REM           LOAD SONG FROM DISK
850   HOME:INPUT"WHICH SONG? ";N$
860   PREFIX$=".D1/"
870   OPEN#2 AS INPUT,N$
875   READ#2;R
880   FOR X=0 TO R-1
900     READ#2;R(X)
910     NEXT X
930   CLOSE#2
950   HOME
955   M=1
960   GOTO 100
1000   IF L<1 OR L>300 THEN L=L:GOTO 100
1010   GOTO 100
```

Pie Man

Picture, if you can, your first day as an untrained baker in a rather large bakery. The product... delicious, creamy pies—the kind your mother used to bake, until she sold out to Conglomerate Bakeries for a fat two hundred thousand.

Your task? To top the pies with whipped cream and a cherry as they are jostled along a conveyor belt. The goodies you need to finish the pies are in bins nearby. Oh, and you must take the completed pies from the conveyor belt and store them in a pie bin, so that credit can be given to you for completing each task. Ah, one slight problem. Should seven pies fall from the conveyor belt and go splaaaaat on the bakery floor, you'll be fired!

A person with great agility could manage quite well, if it weren't for the small obstacles that constantly hinder progress: flour sacks appearing at the most inconvenient times directly in your path; or a grease spot that forces you to drop whatever you happen to be holding, be it whipped cream, cherry or finished pie. And then there's the inebriated baker, who's been nipping at the brandied icing for his wedding cakes; when he stumbles through the bakery uninvited, all forms of havoc erupt.

You would expect more than union wages for such work. Penguin Software only expects you to have fun with this unusual entry into the arcade game market. If you're bored with space games, or tired of being fried by a dozen dragons in a hundred adventure games, then this just might be your slice of cake.

The authors, Eagle Berns and Michael Kosaka, used Mark Pelzarski's Graphic Magician to create the graphics. The game asks which of three modes you wish to use—conventional joystick, Atari joystick (with Joyport or similar interface) or the keyboard. Make your selection and a high-resolution figure of a baker is presented. A conveyor belt begins to move at the bottom of the screen, and a steam whistle lets you know that a baked pie is about to come from the oven for its crowning glory—the whipped cream and cherry.

These condiments are located in bins at the top of the monitor. You move your baker to the whipped cream bin. If you're using a joystick, as I was, press any one of its buttons to collect the calorie-laden white stuff. The baker must now deposit the cream onto the scrumptious pie as it moves along the conveyor belt. The joystick button manages this task, but only if the baker is positioned correctly near the pie. Then it's off to the cherry bin. The baker must then pick up the completed pie with yet another button press, and rush the glorious dessert to the pie bin in the upper right corner.

The first couple of pies are simple. The flour sacks that appear on the bakery floor are easy to maneuver around. Six pies in the bin rate you an apprentice baker.

Now two pies appear almost simultaneously. Always the same routine—first the filling, then the cherry, then a dash to the completed bin. However, a grease spot has appeared on the floor, which, if stepped in, causes your baker to lose whatever he was carrying at that time.

As the game continues, more and more grease spots, more and more sacks of flour, more and more appearances by the drunken wedding-cake baker, leave you precariously close to the limit of destroyed pies you're allowed. Termination waits in the wings. Granted, you've made baker second class, but you've already observed your sixth pie splatter on the floor.

To make matters worse, there are now three unfinished pies moving briskly down the conveyor belt. A feeling of calm desperation sinks in at this point. Knowing full well that three are unmanageable, you try to finish one last pie to better your score before receiving the green slip. Up to the whipped cream. Success! Deposited into the second pie. Avoid three grease spots and flour bags to get the cherry. O.K. Maneuver around the end and put the cherry onto the pie. Pick up the pie. Oh-oh, the lead pie is about to fall. Rush to the bin... and slip in a grease spot as the final pie hits the floor.

Should your score be high enough when terminated, you can enter your initials into the high score file. The five best scores are continuously updated and maintained on the game disk. This added feature really gives you something to shoot for.

Hearty recommendations for Pie Man. Retail price $29.95, from Penguin Software, 830 4th Ave., Geneva, IL 60134. ■

Hartley G. Lesser
***inCider* staff**



International Gran Prix

Fasten your seatbelt... get ready to put the pedal to the metal. International Gran Prix pays your entry fee to a circuit of five internationally known race courses. The cockpit view of a well-clustered instrument panel, and the realistic sights and sounds of quickly moving picket fences, will have you leaning and swaying through every curve. The inevitable spectacular crash will cost time and fuel, but amazingly, will leave you unscathed and anxious to get your RPMs back to top end.

After selecting between Monaco, Riverbank, Oulton Park, Warwick Farm and Karlskoga, you set the number of laps and the track conditions (from icy to vacuum-like) wanted for your race. A Christmas tree starting pole counts down, and you're off.

Paddle 0 is your steering wheel, while button 0 is your accelerator. You have no brakes, but judicious use of the five-speed gearbox, engaged by

quickly releasing and pressing the accelerator button, will get you through even the most difficult "esses."

Keyboard input allows engaging and disengaging cruise control and automatic transmission. Warnings of upcoming curves, such as "paddock," "hairpin" and "esses," appear in time to allow you to ready your nerves for the dangerous maneuvers necessary for you to reach the finish line. A satisfying beep is heard when a new record for the course and condition has been set.

Centermost in the instrument panel of your race car is the tape-line speedometer, which reaches a maximum of 210 miles per hour. Immediately below the speedometer is an indicator which shows wheel direction, and the car's relative position in relation to the width of the track. To the right on the panel is the tachometer, which flashes when a gear's top RPM has been attained. Also situated here is a lap indicator and a fuel gauge. The left side of the panel is dedicated to lap and race times. Idiot lights indicate the use of cruise control, manual transmission, and the sound effects that are in operation.

It's you, the car, the track and the clock. No slow pokes or traffic jams here. No speed limits or citations. Gran Prix is one of the finest simulations this side of NASA.

Gran Prix operates on an Apple II Plus with paddles, 3.2 or 3.3 DOS, or on an Apple III in emulation mode. This racing simulation is from Muse Software, 347 N. Charles St., Baltimore, MD 21201. Price is $29.95. ■

Kevin Burton
Blue Ridge Summit, PA

Shuttle Intercept

Doesn't sound too hard, you think to yourself. Just pick up friendly but dumb satellites with your shuttlecraft and return them to earth with their vital information. At least that's what the instructions say.

You read on. "But watch out!" the instructions say. Be careful of enemy satellites, meteors who don't care who they smash into, missiles and saucers, too!

The cover looks good so you buy the program. You get it home, sit down, and boot the disk. You get paddle 1 in your fingers. You begin play, tense and expectant.

The screen shows excellent graphics. Your shuttlecraft tends to the left side. Ten satellites go by. Into hyperspace you go. Your first ship collides head on with a meteor that comes at you so fast you hardly see it.

Your second ship gets smashed like that too, and you get nervous. Your third ship—things go better. Then the next hyperspace comes on and you get by the meteors. But not the enemy missile. You lose.

That's how Shuttle Intercept for the Apple II can be! Shuttle Intercept was made by Hayden Software's John Van Ryzin. The graphics are good and the sound effects are great.

Four levels of play, with a hyperspace between the first three levels, challenge every user. Your shuttle can take two direct hits, but the third will end the game. You have a proton shield to protect you. If you lose your shield you don't stand a chance. The only one way to survive is to avoid getting hit.

Just one hit drains the antimatter power source and ends the satellite retrieving at Level 4. If you sustain no direct hits after you accomplish level 4, you go back to level 1, and you get your shield back. You don't score points in hyperspace, only in the levels where you collect friendly satellites.

Your weapon is a laser gun. When you push paddle 1's button one time, the shuttlecraft doors open and a hook comes out. This hook grabs the satellites for points. Push the button again and the gun fires.

Enemy flying saucers can't destroy your ship, but they push the grabbing hook back into your shuttlecraft and close the doors. Can't get points like that. You can destroy the saucers and enemy satellites. The only way to spot the difference between an enemy and a friend is to check out their antennas. Meteors cannot be destroyed. Missiles can't be destroyed either, and they track you down and turn you into dust!

Different colors at the bottom of the screen tell you what level you are on, or give hyperspace condition. If your shield took no hits through all the levels, it will go back to level 1. But guided missiles appear on this level and the cycle will continue until you're destroyed and the game is over.

You can get bonus points if you score 5000 points or more. You can interrupt the game to answer the phone or make a snack. You can restart the game if you feel you're doing terribly, and turn the sound off if the baby is sleeping. Different colors indicate the condition of the shield and the ship. I recommend this game for people with some self-control.

Shuttle Intercept is from Hayden Software, 600 Suffolk St., Lowell, MA 01854. Suggested price—$34.95. ■

Kirk Lesser
Hancock, NH

County Fair

Step right up, folks. . . step right up. Try yer hand, right here. Win ol' Teddy here, sir? Yeah, take a chance. . . go ahead. Right here, right here, 'nother winner!" shouts the straw-hatted barker. "One quarter, one measly fourth of a dollar to shoot at this gallery. Make the little lady proud o' yuh."

The atmosphere of the county fair's shooting gallery has been accurately captured by Datamost's software release of County Fair. You can almost smell the roasting of peanuts, taste the sticky sweetness of cotton candy, and hear the screams of the fearless as they ride the Cyclone, while carnival music announces the loading of the game into your Apple computer.

The title page shows you the target types, represented in high-resolution graphics as animals, designs and trees, each target worth from one to five points. Press the button on paddle to clear the title page, and you'll

see the rapid deployment of your bullet supply at the bottom of your monitor.

A handgun appears, pointed toward the quickly appearing targets—white ducks, green and orange rabbits, purple owls, multi-hued balls, twisting, twirling objects, and variously colored trees. Four rows of targets crisscross the screen. Twist your paddle and your weapon changes direction; depress the button to fire with each shot subtracted from the total displayed beneath the gun.

You barely begin the game before realizing that those rotten white ducks won't behave as simple targets. Squawking and fluttering they dip from the bottom row, flying freely back and forth. Shoot them you must, else members of the unruly flock will swoop upon your store of bullets and consume them. Once these ducks begin their feast, nothing you do will save your ammunition.

Infernal rabbits multiply as quickly as you can shoot them, crowding the screen and prohibiting your firing at the trees in the upper portion of the screen. These pines must be cleared if you plan to progress with the game.

Should you miraculously manage to free the screen of moving targets, with bullets to spare, another white duck will show up. But this duck doesn't vanish when shot. It merely changes direction and squawks a lot. The object here is to continue firing at the fowl, preventing it from leaving the screen. Each hit on the duck swells your score in this bonus round, and you'll continue in this mode until you run out of ammunition or the duck escapes.

You will then enter the expert round. Your bullet supply will be reissued, but you won't be smug for long. An apparently identical group of targets is presented, but they're moving at a much accelerated rate. If you think you had problems before, this will test your skills to the utmost.

All in all, County Fair is an enjoyable arcade game. The animation is well done, and the high resolution display depicts each target without color or shape confusion. I was surprised at how much fun I had, as I was drawn into bettering my previous score. The minutes turned to hours as I strove to top 2000 and rate as a true sharpshooter. I'm not there yet. ■

Hartley G. Lesser
inCider staff

Career Directions

Do you like your work? If, overall, you do, chances are that considerable thought, planning and research went into your career choice. Career Directions is designed to help young people with that process.

Not even the most comprehensive and intelligently designed program can replace traditional counseling. But Career Directions, through its interactive sessions with the user, does accomplish two valuable purposes.

By asking many of the right questions, it encourages the user to think realistically about personal strengths, preferences and motivation. How many years of education/training are you willing to invest? In which school subjects have you excelled? Do you prefer to work indoors or outdoors? Alone or in a group? The Career Assessment portion of the program asks the questions, and guides you gently through the self-discovery process.

The Career Analysis portion uses your responses to select a number of appropriate vocational choices. The database comprises 470 occupations, from janitor to biomedical engineer, so any reasonable set of responses will be matched up with some choices. These choices will appear on the screen and will, if you've requested a printout, be provided in a handy list for further exploration.

Are they the best choices?

Well, the assessment validity depends on honest and accurate answers to some penetrating questions. Some youngsters might need help from a more experienced adult before they're ready to formulate good answers—but a run through the assessment section may provide just the impetus needed to seek that help.

Another limitation arises from limited data. The computer's career selections are drawn from a fraction of the possibilities. So, while the range of possibilities is wide enough to be useful, the program is not infallible. A naive user should perhaps be warned to view the computer's selections with a critical eye.

To assist with that critical appraisal, the Career Exploration portion of the program provides detailed information about any of the included job categories. The documentation (which, incidentally, is helpful and intelligent) suggests that after the first run you work back and forth through assessment, analysis

and exploration to refine the choices.

Its limitations noted, I suggest that Career Directions also has several advantages over personal counseling. The computer is blind. It doesn't know your age, race or gender, and thus can't be subtly (or blatantly) biased by irrelevancies. And, as with all computer applications, the program is infinitely patient—it won't greet your 17th career reversal with wild-eyed exasperation.

Career Directions was programmed by Julie Margolis, and is distributed by System Design Associates Inc., Suite 403, Union Building, 723 Kanawha Blvd. East, Charleston, WV 25301. Program and database disks are provided; a single disk drive can be used. The Career Directions package costs $59.95. ■

Linda Stephenson
***inCider* staff**

Ceiling Zero

When Turnkey Software states in their instruction booklet that Ceiling Zero is a high-speed action game, you'd be hard-pressed to accuse them of misinformation. This arcade game, written in machine code, is extremely rapid, and is another outstanding refinement of the original space invader class of software.

This game's difference is enjoyment, of which you'll find a great deal if you can also tolerate a high level of frustration. High-resolution graphics and sound add to the total flavor of Ceiling Zero, making it an outstanding addition to your Apple game library.

The title page reveals the points awarded for shooting a variety of aliens. The game begins when you press paddle 0's button. A noisy mother ship descends onto your screen, followed by two disk-shaped objects that resemble microwave transmitters. These position themselves at each side of the upper screen, and emit a beam, a force field of some type, which materializes between the two to form the ceiling. This is ceiling 15, the highest and simplest of the playing modes. From this point onward, the ceiling will lower each time you successfully defeat an alien attack wave.

You begin the game with three bases, and a glance at the scoring portion of the screen will tell you how many remain throughout play. Once in play, the alien mother ship flies to and from above the ceiling, dropping the annoying and destructive aliens from her hold. Their one mission is to eradicate you, which they achieve by dropping on your base and blowing you to Kingdom Come.

Should the aliens miss you, they'll bounce like rubber balls between the ground and the ceiling until you zap them into oblivion, or they you. Each "ceiling" becomes progressively more difficult. The nasty aliens get quicker and more of them converge to enliven your game.

Not only aliens threaten—the mother ship commits landing craft to the fray too. Should one of these vessels land, your base is immediately wiped out. One of your more difficult tasks is to destroy the landing craft while dodging the dreaded aliens. But then again, you don't buy a game for ease of play, but to be challenged.

The author, Stephen Warady, has brought a fresh approach to the alien shoot-em-up arcade game, and has done well. With 15 levels of difficulty, and high-speed action, you may well climb your own ceiling before mastering Ceiling Zero.

Ceiling Zero, from Turnkey Software, 13078 Mindanao Way, Suite 314, Marina Del Rey, CA 90291, costs $29.95. ■

Hartley G. Lesser
***inCider* staff**

Apple Basic for Business For the Apple II

by Alan J. Parker and John F. Stewart

Reston Publishing, a Prentice-Hall Company
Reston, VA 22090

Let's call him Jim, a fictitious businessman who's decided to computerize his small hardware store. He reads all the literature he can get his hands on, subscribes to *Microcomputing*, asks around, and finally buys an Apple II.

At the same time, he picks up a copy of *Apple Basic for Business: For the Apple II.* It seems more interesting and should have more workable, practical examples than the instruction manuals that came with Jim's Apple.

After all, the preface of the book notes that, ". . . as we look at the many facets of a problem and introduce the necessary Basic statements, the student will collect the tools needed to solve all these problems." This sounds darn good to Jim. So does the authors' promise that, "On the successful completion of this text alone. . . the student will understand the fundamentals of programming and be able to write programs of reasonable complexity."

Jim quickly learns how to start his Apple II with its disk, and writes the first couple of short programs listed in the book. Not bad—they work! But what's this? On page 17, the problem summary calls for an invoice program—you're to multiply 50 units sold by $15 per unit, which should give you $750 as an answer. But the program listed is the Inventory program from the next page. This is confusing enough, but to make matters worse, the *incorrect* program gives the *correct* answer!

Not possible, you say? But gee, on page 18 the authors claim that, "All program listings in this book were produced on a line printer for clarity." That can't be right—here they show a program listing, list it, type Run. . . and it gives the answer to another program!

Well, okay, Jim thinks, must be a typographical error, and anyone can have one of those. . . .

Ah, here's that same inventory program on the next page, where they want to start with 120 units in stock, receive 40 and sell 45 to end up with 115 left. The same program listing, another Run—only this time it produces the proper answer of 115.

A number of short programs are now used as examples and Jim dutifully types them in and runs them, but now he notices something else. To ensure a program has "finished as it should," the book tells Jim to use a Stop statement, so the screen will indicate BREAK IN______, with the line where you stopped shown. Well, sure enough, things do stop there, but who needs the Apple to beep at you each time? No major problem, but it sure is annoying! Wouldn't it be better, Jim wonders, to use as your last line, 500 PRINT "END OF PROGRAM": END. Now, you'll know it's finished without all the noise.

By page 32 Jim is working with a decent inventory program, with five fields printed in a nice row by his printer. Unfortunately, there's not even a mention in the book that printers and other devices have to be accessed by the Apple II, nor is there any mention as to how you might do such things. Obviously, each device has its own code, so there's no way a book could instruct someone on all of them, but it would have been nice if the authors had at least said *something* about controlling the printer.

Other Little Things

Jim read in the Apple II manual that Let is an optional statement, that 100 LET A = 55 is the same as 100 A = 55. In this book Let is described on page 35, and it's another 54 pages before Jim reads that it's optional. List is explained, but not control-S, which stops a listing—now that the programs are getting longer, they flash by so fast that Jim can't keep up with them.

Jim makes a few errors reading things because zero is printed as 0, rather than as Ø. Good advice on page 13 tells Jim not to use fields such as I1, as it's hard to distinguish between the I and the 1, and then later on the book asks you to name a field I11. Jim wonders if anyone read his own advice.

Well, it could be worse. A typographical misprint in a program, a few minor annoyances.

But things don't get any better.

By now Jim's working in chapter 3, learning how to input data from the keyboard. The flowchart, as usual, is readable, but the program asks,

```
120 PRINT "TYPE NAME, HOURLY RATE,
REGULAR HOURS, OVERTIME HOURS"
130 INPUT N$, R, H1, H2
```

Now, Jim's got to remember not only *what* to input but also in *what order* to do it. Hmm. . . . let's skim ahead, to a more complex problem on page 68. Here they ask our businessman to input six different things, all in a row. When he's done, he's told to type, "99, 99, AA, 99, 99, 99." This is *easy* computer work? Time-saving? Typing six strange data items just to tell the Apple, "I'm done"?

Is it really a problem to handle things differently, so they're easier to do?

Well, you decide.

The program for payroll computation on page 68 reads this way:

```
100 PRINT CHR$(4) ; "OPEN EMPLOY"
110 PRINT " TYPE EMPLOYEE NUMBER,
DEPARTMENT NUMBER, EMPLOYEE
NAME"
120 PRINT "HOURLY RATE, REGULAR
HOURS"
130 PRINT "OVERTIME HOURS SEPA-
RATED BY COMMAS"
140 PRINT "WHEN FINISHED TYPE 99,
99, AA, 99, 99, 99"
150 INPUT N, D, N$, H, R, V
160 IF N = 99 THEN 400
```

Line 400 will close the file.

Jim spent some more time figuring a better way:

```
110 INPUT "EMPLOYEE NUMBER (AN-
SWER -1 TO END) ";N
115 IF N = -1 THEN 400
120 PRINT: INPUT "DEPARTMENT
NUMBER";D
130 PRINT: INPUT "EMPLOYEE NAME
"; N$
140 PRINT: INPUT "HOURLY RATE";H
150 PRINT: INPUT "REGULAR HOURS"
```

RH ELECTRONICS, INC.

COPYRIGHT © 1981 — PATENTS PENDING
566 Irelan, Buellton, CA 93427
(805) 688-2047
8:00 TO 5:00 CALIFORNIA TIME

COMPUTER PRODUCTS
DESIGNING • MANUFACTURING
ELECTRONIC ENGINEERING

SUPER FAN II™

FOR YOUR APPLE II* COMPUTER

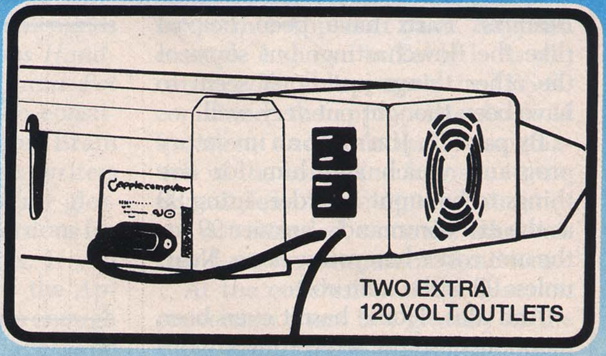


One Year Warranty

$74.95

With Zener Ray™ Protection $109.00

MASTERCARD — VISA



"COOL IT"

- ALSO FITS ON APPLE'S* NEW MONITOR STAND
- RED PILOT LIGHT ON/OFF SYSTEM SWITCH
- CLIPS ON — NO HOLES OR SCREWS • REPLACEABLE SWITCH
- AVAILABLE IN 120V or 240V AND 50/60 HZ • DURABLE MOTOR
- REDUCES HEAT CAUSED BY EXTRA PLUG-IN CARDS
- SOLD WORLD WIDE • UNIQUE 1 YEAR WARRANTY
- TAN OR BLACK COLOR • QUIETEST FAN ON THE MARKET
- INCREASED RELIABILITY — SAVES DOWN TIME AND REPAIR CHARGES
- LOW NOISE DUE TO DRAWING EFFECT OF AIR THROUGH YOUR COMPUTER AND SPECIAL FAN AND MOTOR DESIGN
- TWO EXTRA 120V OUTLETS FOR MONITOR AND ACCESSORIES TURN ON WHEN YOU TURN ON YOUR FAN (NOT AVAILABLE ON 240V MODEL)

SUPER FAN II™ WITH ZENER RAY OPTION $109.00

ZENER RAY™ TRANSIENT VOLTAGE SUPPRESSOR

OUR BUILT IN ADVANCED DESIGN UNIT GIVES
DRAMATIC COST SAVINGS — STOPS ANNOYING DOWN TIME
INSURANCE FROM VOLTAGE SPIKES - GLITCHES
DANGEROUS VOLTAGE SPIKES CAN JEOPARDIZE YOUR COMPUTER SYSTEMS
PROTECT COMPUTER - DISK DRIVE - PRINTER AND MONITOR

NO CUTTING WIRES • WON'T VOID WARRANTY, JUST PLUG IN SUPERFAN II WITH ZENER RAY

OTHER PRODUCTS BY RH ELECTRONICS, INC.

SUPER RAM II™ 16K RAM CARD FOR YOUR APPLE II. 2 YEAR WARRANTY $125
GUARDIAN ANGEL™ AN UNINTERRUPTABLE POWER SOURCE $595
12 VOLT TRANSVERTER 12 VOLT — RUNS YOUR APPLE II COMPUTER AND AND 5¼" DRIVE FROM YOUR CIGARETTE LIGHTER $149

*Registered trademarks of Apple Computer Inc.

DEALER INQUIRIES INVITED

```
;R
160 PRINT: INPUT "OVERTIME
HOURS";V
```

Jim discovers this new program's done a couple of things. First, it's much easier to read and work through—you enter one number at a time, as your terminal is requesting only one piece of information at a time. Much easier to handle in the real world.

Also, there's spacing between the questions (the print statements) so things are easier to read. To finish, you type one thing (– 1) not 6. And finally, Jim put the PRINT CHR$(4); "OPEN EMPLOY" at the *end*, after the data was input, as there's no reason to open a file, turn on the disk, then input data, and turn on the disk again to save it. Why not do it one time and save the wear and tear on the disk motor?

By now Jim doesn't know quite what to think about *Apple Basic for Business*. Parts have been helpful (like the flowcharting) but some of the other things just don't seem to have been thought out very well.

By page 73, Jim's into an inventory program, which asks him for five things to be input in order, using 99 as the exit command—answer 99 for the unit cost when you're done. Neat, unless 99 *is* the item cost.

And alas, Home hasn't even been mentioned, so the screen quickly fills up with all kinds of stuff, making it hard to read and see what's going on. Jim checks the index. . . Home isn't even listed.

Files

Now, Jim's been thinking about his own computer programs, and seeing that *Apple Basic for Business* has four chapters on sequential (text) files and only one on random access files. He figures that sequential (text) files must be the way to go.

But as he reads through these chapters, he discovers that in order to change data—and business data in the real world is constantly changing—in a sequential file, you have to (1) change exactly the right number of data characters, (2) add data to the end of a file, or (3) recopy the entire file, changing items as you go along. It uses disk space efficiently, but sure can be slow.

Well, Jim got to thinking he really didn't care if a specific job took two disks or four. What concerned him most was speed and data access when he needed the information. So he found that the chapters on sequential files were good but not of much help.

He also found some good points. Authors Parker and Stewart give a brief example of displaying what's happening inside a program on the screen. This is valuable for a user, not only to see what's going on with the data flow, but also to have some idea time-wise of how far along a run is. They also have some good advice on breaking data down into various usable categories—not just presenting total payroll, for instance, but perhaps printing it out by department, etc.

"In Jim's real world, things don't happen that way."

The chapter on arrays is effective, with its examples using tax tables serving as good instruction and as something Jim might use in his own payroll program.

Both Parker and Stewart are associate professors; this gets in their way when the example inventory program gets going strong. They suggest that if you don't have enough of an item on hand to fill an order, you simply indicate that your stock isn't sufficient and cancel the customer's order.

In Jim's real world, things don't happen that way. You sell what you have in stock and *back-order* the rest of the order, but you darn well sell something.

Also, as Jim tries this program, he notices that it requests three data inputs in a row, and discovers another disadvantage of handling data this way. Once you input it like that, you must write the data to the disk in this way:

```
565 PRINT A1;",";A2;",";A3
```

That's an awful lot of quotation marks and commas that have to be in exactly the right places. Jim tries it once, twice, and makes too many mistakes. So, he changes the program to this:

```
565 INPUT "QUANTITY"; A1
575 PRINT: INPUT "COST"; A2
585 PRINT: INPUT "STOCK NUMBER"; A3
. . . . (set up the disk file)
600 PRINT A1: PRINT A2: PRINT A3
```

Easier to write, easier to read, easier to understand. And once again, Jim wonders what he bought this book for. He's doing all the work.

Jim sighs and continues reading. He reads a brief discussion on canned programs, structured (modular) programming, editing and graphics.

The Heart of the Problem

Then Parker and Stewart insert a Printusing routine, and this, Jim quickly decides, exemplifies what's wrong with this book.

First, they use it as a subroutine—Gosub 6000—but without any mention so far of the Gosub function. What's this?

Jim thumbs through his Apple II manuals and finds out about Gosub. Kind of neat—you write the routine once and you can use it over and over. The ability to do something that powerful—why, that's a major strength of the computer. So, Jim searches back through *Apple Basic for Business*, but he can't find Gosub. He does notice that early on Parker and Stewart say that "language capabilities that do not fit clearly into a problem context were omitted for the purposes of this text."

"Ah, they warned me," Jim mutters.

But to leave out something as powerful as Gosub means something is wrong with their problems, not with the language. Parker and Stewart should have restructured their problems and programs to in-

clude and make wide use of the Gosub function, rather than just ignoring it because it didn't fit their particular problems.

But darn, Jim thinks, Gosub has to be here someplace, so he flips ahead in the book. Sure enough, a couple of pages away a whole paragraph is devoted to Gosub.

But again, you be the judge: would the first-time computer user be able to "understand the fundamentals of programming and be able to write programs of reasonable complexity" after reading this explanation of Gosub:

> "The GOSUB 6000 is used to 'call' the subroutine. GOSUB is similar to GO TO except that the computer remembers in what line the subroutine was 'called' and returns to the statement following the 'calling' line when the subroutine finishes with a RETURN statement..."

Jim wonders, what's a subroutine? Since he can't find the term in the index, he checks around. Ah, here it is: "A subroutine is a section of a program that is needed in several different parts of the program. Instead of repeating it, it is entered once and then 'called' whenever it is needed."

Is this relationship between Gosub/Return and their subroutine(s) clear to everyone now?

To make things even worse, this Printusing routine uses MID$tring and LENgth functions that haven't been explained and never are.

And so, following their brief (and poor) explanation of what a subroutine can do and how Gosub works, Parker and Stewart ask you to insert this into your own programs, without understanding how all the Basic functions inside the subroutine work, without even understanding how the subroutine itself works.

Jim pushes himself through the last part of the book. At the end, Parker and Stewart put in a Summary of Basic Commands, which perhaps is more notable for what they left out. These (among others) are missing: Flash, Inverse, Normal, Trace, NoTrace, Home, Clear, Gosub, Read, Restore and Stop.

So Jim does. He programs himself to get out the manuals that came with his Apple II—and "files" *Apple Basic for Business*. ■

Gregory R. Glau
Prescott, AZ

The Basic Conversions Handbook for Apple, TRS-80, and Pet Users

by David Brain, *et al*

Hayden Book Company, 1981
50 Essex St.
Rochelle Park, NJ 07662
Softcover, $7.95

Found a TRS-80 program you'd like to run on your Apple, but haven't got the slightest idea of how to convert the Basic code into something compatible with your system? Oh, sure, some of the TRS-80 Basic seems similar, but then again... there are those SET(x,y) statements and strange STRING$ commands that baffle immediate explanation.

What you need to get your hands on is *The Basic Conversions Handbook for Apple, TRS-80, and Pet Users*, by four authors who constitute a company called the Brain Bank. In 79 pages, these writers manage to present, in several glossaries, programming conversions for the listed computer systems. If you happen to be enamored of the Apple, this book has more than enough information presented to get you cracking converting TRS-80 (Model I or III) and PET Basic programs for your computer.

The preface briefly explains why the book was written, and gives the reader further impetus for delving into the succeeding pages to translate programs. You're also advised that the conversion of video graphics is basically left to one's imagination, as the formats differ from machine to machine. However, if graphics are of real interest to you, each chapter dealing with specific computer conversions discusses the graphics capabilities of that particular system. (See also the article "Apple/80" on p. 96.)

The reader is also advised to fully understand the actions of If...then statements, as these could cause major glitches in a conversion if translated incorrectly.

Chapter 1 talks about the methods of translation for the Basic language, and advises the reader to initially figure out the flow of his program before attempting the translation. This advice is excellent—your conversion will take far less time, and graphics translations will be far easier, if the text and/or video screens have been charted beforehand.

Machine-language programs are not covered in this book. Each chapter deals with only one given language. The reader is shown a sample conversion of a TRS-80 line listing to Applesoft. This simple conversion is offered to give you confidence. Clearing the screen, saying "Hello," and then ending the program is no problem.

Probably of most interest is chapter 3, where TRS-80 language is converted into Applesoft Basic. A typical example is shown in Table 1. As you can see, the table presents the TRS-80 command, the appropriate Apple II function, and a necessary explanation of the command to make its use crystal clear. Some Apple commands are not included, since the other micros have no comparable functions.

At the conclusion of the function conversion chart is a glossary of the type declaration characters ($ = string = A$,F$), arithmetic operators, string operators and relational operators, and a display of the order of operation.

Conversion of Apple II programs into TRS-80 Basic is covered in Chapter 2. The commands are presented in alphabetical order, with the Apple function described first, followed by the TRS-80 function, and any comments regarding the particular command ending the presentation. Approximations of the Apple Peek, Poke and Call statements are listed, as well as Apple Pokes you can ignore when translating for the TRS-80.

Each chapter includes specific information for PET conversions, with the PET dominating chapter 4 for TRS-80 and Apple Basic translations.

Useful subroutines, all ready for

the reader to enter into his or her system, fill Appendix A, with subroutines such as string storage, print using function, string recall and convert, STRING$ function and random function.

Sample program conversions, found in Appendix B, are most enlightening, and reveal how easy the handbook is to use. Charts of screen coordinates and screen memory locations, to assist in screen formatting, are located in Appendix C.

I recommend this handbook to the novice or advanced computerphile. For the latter, having the necessary commands close at hand saves time and expedites translation. For the former, this handy reference not only teaches your chosen Basic, but shows how Basics relate. Entirely new worlds of computer software will open up to you.

Now, where did I put that copy of *80 Micro*? I need that touch-typing tutor program listing for my Apple. ■

Hartley G. Lesser
***inCider* staff**

An Introduction to Microcomputers Volume 0, The Beginner's Book (3rd Edition)

by Adam Osborne and David Bunnell

Osborne/McGraw-Hill 1982
Berkeley, CA
Softcover, 240 pp.

Anyone curious about the microcomputer boom, anyone yearning to buy a micro, is sure to find at least one item of interest here. Don't leaf too quickly through the book—Osborne and Bunnell cover so much ground, you might get dizzy. Although used too lavishly in spots, the bold-face highlighting of key thoughts is a helpful guide.

The first half of the book sets out a history of microcomputers, introduces us to basic hardware, whisks us through the maze of micros on the market, entertains with a tale of small-business micro acquisition, and offers a tantalizing glimpse of software.

Most timely is the discussion on choosing a microcomputer for business applications. Although the authors refer the reader to another book of theirs devoted entirely to this topic, the information here would be more useful in a systematic format, using parallel criteria for comparison of various brands and types of micros.

The micro evaluation segment is nonetheless lively and to the point. "The Video Display," for example, points up effectively the key considerations in selecting a video monitor. From page 23: "A vital distinction among microcomputers is how they relate to their video displays, whether they are monitors or television sets. *Some monitors will only display uppercase letters;* others will display both upper- and lowercase. This plus the number of characters displayed per line (usually 40 to 80) are very important considerations for word processing and other applications where the end result is letters and numbers printed on paper." The subsequent discussion of video text-scrolling methods is enhanced by well-executed graphics.

Pros and cons of micros of various prices and capabilities are set out in a fairly straightforward manner for the less widely used micros. But for the big sellers—Apple, Atari, Commodore, Radio Shack—the balance swings toward historical importance rather than current advantages. This leads me to wonder whether the authors feel their popularity owes mainly to their positions as "firsts."

The volume's second half gets down to binary and Boole, disk storage and digital signals. It does a good job of packing in lots of technical information to take the patient reader well into the bits-and-bytes level. Although elaborating the binary-octal-hex conversion and arithmetic section a bit out of proportion, Osborne and Bunnell have saved the eager-to-learn reader from the hunt for other sources by skillfully fitting a lot of useful information between two covers. ■

Virginia Schaefer
Boston, MA

by Lee E. Sumner, Jr.

The Mad Slasher, and More...

Once I had learned the method by which I could draw lines on the Apple screen, there was little that could draw me away from such activity. After continuous experimentation with Apple's high-resolution graphics, I ended up with a computer full of wonderful, useless one liners.

A "One Liner" is a program that is exactly one Basic line in length. The "One Liner" requires no initialization, and repeats to form some kind of graphic display. The programs as revealed in the listings were written purely for the fun of doing them. It really is a challenge to pack as much action as possible into a single line of code.

```
1  HGR : POKE  - 16302,0: FOR Q = 0 TO 100: HCOLOR=  RND (1) * 7 + 1: HPLOT
     RND (1) * 279, RND (1) * 191 TO  RND (1) * 279, RND (1) * 191: NEXT
    : FOR X = 1 TO 3000: NEXT : GOTO 1
2  HGR : POKE  - 16302,0: HCOLOR=  RND (1) * 7 + 1: HPLOT  RND (1) * 279, RND
    (1) * 191: FOR Q = 0 TO 100: HPLOT  TO  RND (1) * 279, RND (1) * 191:
     NEXT : FOR X = 1 TO 3000: NEXT : GOTO 2
3  HGR : POKE  - 16302,0: LET Z =  RND (1) * 9 + 4: HCOLOR=  RND (1) * 7 +
    1: FOR Q = 0 TO 279 STEP Z: HPLOT 140,191 TO Q,0: HPLOT 140,0 TO Q,19
    1: HPLOT 0,96 TO 279,Q / 1.46: HPLOT 279,96 TO 0,Q / 1.46: NEXT : FOR
    X = 1 TO 3000: NEXT : GOTO 3
4  HGR : POKE  - 16302,0: LET Z =  RND (1) * 9 + 4: HCOLOR=  RND (1) * 7 +
    1: FOR Q = 0 TO 279 STEP Z: HPLOT 0,191 TO Q,0: HPLOT 279,0 TO Q,191:
     HPLOT 0,0 TO 279,Q / 1.46: HPLOT 279,191 TO 0,Q / 1.46: NEXT : FOR X
     = 1 TO 3000: NEXT : GOTO 4
5  HGR : POKE  - 16302,0: LET Z =  RND (1) * 9 + 4: HCOLOR=  RND (1) * 7 +
    1: FOR Q = 0 TO 279 STEP Z: HPLOT 70,96 TO Q,0: HPLOT 210,96 TO Q,191
    : HPLOT 140,48 TO 279,Q / 1.46: HPLOT 140,144 TO 0,Q / 1.46: NEXT : FOR
    X = 1 TO 3000: NEXT : GOTO 5
6  HGR : POKE  - 16302,0: LET Z =  RND (1) * 9 + 4: HCOLOR=  RND (1) * 7 +
    1: FOR Q = 0 TO 279 STEP Z: HPLOT 140,144 TO Q,0: HPLOT 140,48 TO Q,1
    91: HPLOT 70,96 TO 279,Q / 1.46: HPLOT 210,96 TO 0,Q / 1.46: NEXT : FOR
    X = 1 TO 3000: NEXT : GOTO 5
7  HGR : POKE  - 16302,0: LET Z =  RND (1) * 9 + 4: HCOLOR=  RND (1) * 7 +
    1: FOR Q = 0 TO 279 STEP Z: HPLOT 70,144 TO Q,0: HPLOT 210,48 TO Q,19
    1: HPLOT 70,48 TO 279,Q / 1.46: HPLOT 210,144 TO 0,Q / 1.46: NEXT : FOR
    X = 1 TO 3000: NEXT : GOTO 5

]PR#0
```

Seven one liners for action-packed graphics.

The programs are listed as though one program. To call a specific one liner, merely enter

RUN n

where *n* is the line of the desired program. Each program will continue to run until halted by either a control-C break or the total destruct key, reset. When you enter these programs, don't use any spaces.

Program 1 I call the Mad Slasher. Not much order, but plenty of action. Program 2 simulates a superball bouncing off invisible walls.

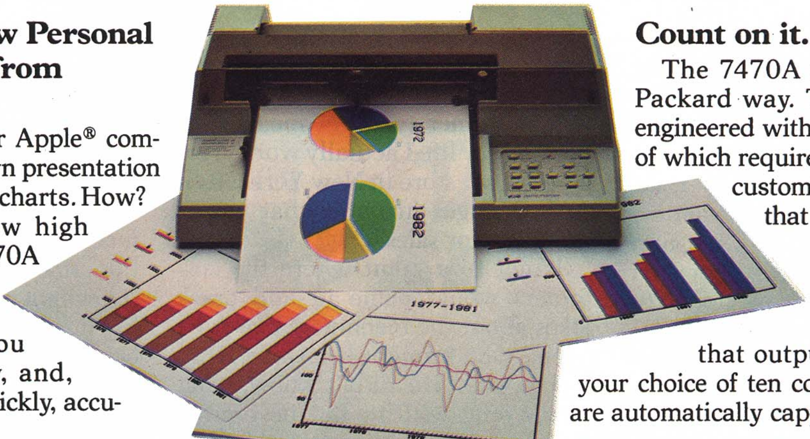
In programs 3 through 7, the constant 1.46 is the ratio of 280 to 192. This allows you to step both the x and y axes with only one For... Next loop. You can create your own patterns by changing the beginning part of each HPLOT statement. If you initially plot the locations I used on a piece of graph paper, I think you'll see the method I employed.

I use these programs as demonstrations of the speed and ease of use inherent in Apple graphics. A library of these programs can give you some real "braggin" rights when your TRS-80 friends are around! ■

Address correspondence to Lee E. Sumner, Jr., 75 E. King St., Dallastown, PA 17313.

One Apple® and $1,550 can make a lot of pies.

And charts. And graphs.



Introducing the New Personal Computer Plotter from Hewlett-Packard

Now you can use your Apple® computer to generate your own presentation charts, graphs, and pie charts. How? Simply add on the new high quality, low cost HP 7470A Personal Computer Plotter.

The 7470A helps you save time, save money, and, lets you communicate quickly, accurately and effectively.

The eye is faster.

Data, when visualized graphically, becomes information fast. Charts and bar graphs can make any presentation clearer and more readily understood. But asking your staff to produce the graphics for your next presentation doesn't ensure accuracy or artistic talent. And going to outside suppliers can be costly. Combined with your Apple® computer, the new HP 7470A plotter does the communicating for you. Quickly. Logically. And with off-the-shelf software.



Fast and pretty.

The 7470A gives you high plotting speed with excellent line quality...faster than any competitive small plotter. On top of all that, it comes in an attractive design package that looks nice on your desk. And it does it for only $1,550. (U.S.A. domestic suggested retail price.)

Count on it.

The 7470A is built the Hewlett-Packard way. To last. Designed and engineered with only a few parts, none of which require adjustment. And with customized integrated circuits that ensure reliability.

Pen Pals.

The HP 7470A has two single-pen stables that output multi-color plots in your choice of ten coordinated colors. Pens are automatically capped and stored.

An option you'll want, too.

For only $95, you can also get a 17057 Overhead Transparency Kit that turns your plots into transparencies for overhead projectors. For "I need it tomorrow at 9:00 A.M!" meetings, it's a necessity.

Start plotting your next presentation today. Clip and mail the coupon below. Now.

Mail the coupon below and we'll send you—absolutely free—a sample plot, a more detailed brochure, and a sample overhead transparency.

Then...stop in at your nearest Hewlett-Packard Dealer. See the HP 7470A in action. Once you see it demonstrated you'll find a hundred ways to make your own Apple® pies. And charts. And graphs.

When performance must be measured by results

HEWLETT PACKARD

Seeing is believing. Send me a sample plot, an overhead transparency, and more detailed information.

Name ______________________ Title ______________________

Company ______________________

Address ______________________

City, State & Zip ______________________

Phone Number (   ) ______________________

Send to: Hewlett-Packard, 16399 W. Bernardo Drive, San Diego, CA 92127—Attn: Nancy Carter

by Bill O'Brien

A Fresh Start

If you expected Suzanne Somers, I can almost guarantee you're going to be disappointed (I never did look good in a wig).

Your reading this column right now (or even later, for that matter) tells me that you probably fall into one of two categories: either you own an Apple III, or you don't (to this day, I'm grateful to the Axel O'Malley School for Deductive Reasoning). In either case, I'm willing to bet you have a lot of curiosity about what's happening for, with and to them. This is the first of a series, and it may be a little different than most. I'd like to see if it's possible to have serious discussions about the care, feeding and use of a computer and not destroy the fun of owning one.

Why a Henway?

A while back somebody asked that fateful question. The same goes now: Why an Apple III? Why not any of the other names found in the litany of computers that assails our consumer senses every time the urge to join the electronic revolution sets in?

How about a very simple answer: "I bought the Henway because it was a Henway and at the time, I wanted a Henway."

I can hear you all now: "What the heck is this guy talking about? I thought this was a computer column? What's a Henway?" (About two pounds—I couldn't pass that up.) The point is, you really shouldn't buy something just for a name (or *not* for a name, for that matter).

Apple got a lot of very bad publicity when it introduced the III in 1980. The first groups of machines didn't work well. I had one that liked to reboot itself, or just crap out, usually as I was trying to show it to a customer (you got me; at one point in my life I actually worked for a computer store in New York City). As you might imagine, that wasn't too great for sales.

It got to a point where the customers would come in and crack jokes about the machine ("I've got a friend who brought his Apple III in for service. He didn't know what was wrong with it, but he figured there must be something—it worked.") They were right. Apple knew, I knew, and they knew it. It was a bad time for the III.

But Apple did something that was surprising. They didn't just bail out and throw the III to the dogs. They dedicated more people and more money to finding out where they had gone wrong. A lot of it really was hardware problems, and some of it was dealer problems.

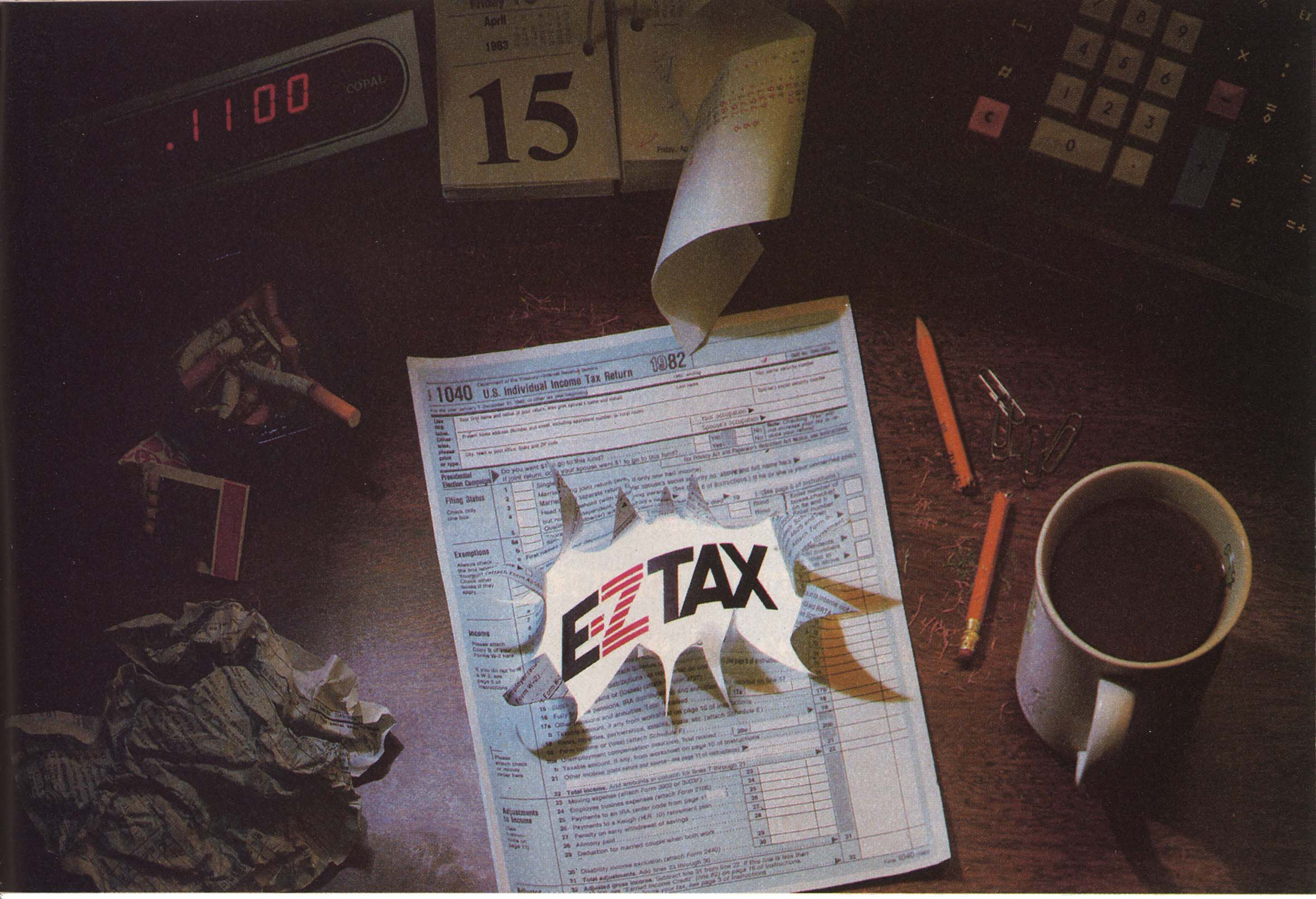
The Good, the Bad, the Unexpected

The Apple III is a sophisticated machine. Before the III's introduction, many dealers thought that configuring a computer meant unpacking it, plugging it in along with any boards it needed, and turning it on. Show and Sell. Now they had a machine that had to be learned. The operating system had to be set up for the number of disk drives and the type of printer used. Basic was different from the II's Basic. There were no built-in languages—everything was loaded in from disk. Even the manuals were different. They weren't as friendly as the Apple II manuals, and no one seemed to be

```
 10  HOME
 20  VPOS=5
 30  HPOS=1
 40  INPUT"What is your name ?",NAME$
 50  HOME
 60  VPOS=5
 70  HPOS=1
 80  PRINT"Hi there, ";NAME$
 90  PRINT"Do me a favor, ";NAME$
100  PRINT"Would you please press the 'Y' key ?"
110  GET KEY$
120  VPOS=10
130  HPOS=1
140  PRINT"Thanks, ";NAME$
150  IF KEY$<>"Y" AND KEY$<>"y" THEN 200
160  PRINT"That was really nice of you..."
170  GOTO 300
200  PRINT"But I really wanted the 'Y' key"
210  PRINT"Want to learn the alphabet and"
220  PRINT"come back later."
300  END
```

Program listing. Apple III Business Basic sample program.

Address correspondence to Bill O'Brien, 11 Dongan Place, New York, NY 10040.



The Tax Break You've Been Looking For!

You Just Found It!

E-Z Tax — the easiest to use tax preparation software ever developed. Now you can do your own tax return without any knowledge of taxes or computer programming.

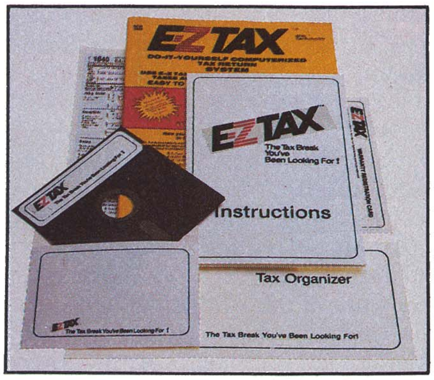
E-Z Tax's self-prompting questions assure you that nothing is overlooked. The program automatically computes the lowest tax for you.

E-Z Tax prepares the following IRS Forms and Schedules

1040 A	2106
1040 EZ	2119
1040 P. 1 & 2	2210
Sch. A	2440
Sch. B	2441
Sch. C	3468
Sch. D	3903
Sch. E	4137
Sch. F	4684
Sch. G	4972
Sch. R/RP	5695
Sch. W	6251
1040 SE	6252
1040 ES	

Prints on Federal Forms

It's so easy to use, you'll be doing your tax return the moment you insert the 5¼" disk. When you're finished E-Z Tax will print out your information on the official Federal Forms.



Very Friendly!

E-Z Tax is user friendly. When you have a question, the program will tell you on what page in the E-Z Tax Guide Book you'll find the answer. If you make a mistake the program alerts you immediately with screen prompts.

We Support You!

E-Z Tax has a toll-free customer service number available 24 hours a day for your convenience.

Never in the history of computer software has there been so much for so little.

ONLY **$69.95**
TAX DEDUCTIBLE

Available in the following versions. Apple II (48k), IBM PC (64k), Atari and CP/M.

For the name of your nearest distributor **CALL:**

408/264-1040
800/331-1040 U.S.A.
800/344-1040 CA

E-Z TAX T.M.

2444 Moorpark, San Jose, CA 95128

Apple II™ is a trademark of Apple Computer, Inc.
CP/M™ is a trademark of Digital Research, Inc.
IBM™ is a trademark of IBM Corp.
Atari™ is a trademark of Atari, Inc.

Circle 155 on Reader Service card.

able to answer questions.

When a computer has an erratic personality and needs time devoted to it, and the two occur at the same time, you find that many times it violates the dealer's laws of profitability. And when that happens, not many machines are sold.

Apple upgraded the hardware so that everybody buying the newer machines would have reliable models. Then they did something that will forever endear them to my heart: they swapped new machines for old. Anyone who had purchased an Apple III with a serial number below 14000 was given a new one in exchange.

Dealers often received Apple III technical tips and answers to the questions people were asking. A year after its debut, the III had its second coming (portentious words, if ever I've heard them). That's one reason I bought the III: Apple really was (and is, so it seems) committed to the product.

Personal Considerations

But that was only one reason. When you go out looking for a computer in the same class as the Apple III, you're forced to make certain assumptions (oh, but I hate that word). We're going to assume that we'll leave out everything that doesn't have an 80-column screen. Why an 80-column screen? Convenience.

You buy a computer so that, in some ways, you can be lazy. Telecommunications systems are usually more useful with 80 columns of printing available. Word processing programs emulate the printed page, which, coincidentally, is ruled for 80 columns of characters. Switching screens, sliding back and forth, they all might be passable in a hobby or casual use situation, but a lot of 80-column boards are sold for Apple II's. Spreadsheet applications are running optimally when the greatest number of columns of data can be viewed at the same time.

We're also going to exclude anything that doesn't have upper- and lowercase letters. Let's face it, we learned all that big letter and little letter stuff in grade school. We spent a lot of time copying them into our notebooks and writing them over and over again for homework. Why waste all that effort and energy? Let's deal with lowercase—it's a recognized part of the language.

A Peripheral Choice

Disk drives are another thing to consider. We're going to eliminate any computer companies that don't supply disk drives with their own brand name. There are times when you buy a computer from this company and an add-on disk drive from another. They don't work together for some reason. Who takes responsibility? Is it the computer that doesn't work with the add-on, or vice versa?

Not every company is going to say, "Oh, it's not our part, it's that other one." That would be as unrealistic as believing that every company would, or could, take responsibility for *every* application the product was used in. Sometimes it is easier to be able to say, "Apple, I've got an Apple drive connected to my Apple computer with an Apple cable, and it doesn't work. How come?"

Also, some applications require a lot of storage, more than you could find on most of the floppy disks available. For these uses, you might require a hard disk, which has much more memory. While there are companies that offer them in varying sizes for several different machines (including Apple IIIs), Apple, among a few others, markets their own drives (made by someone else, but carrying the Apple label).

Apple also has a relatively low-cost service agreement for complete carry-in repairs, and a heck of a lot of authorized Apple dealers (and every one of those little guys is an authorized Level 1 service center). While it's not unheard of, it is rare that a machine or part has to leave the dealer to be repaired.

The Disclaimer

Now please, let's not flood the post office with letters from manufacturers of Apple III compatible hardware. I'm not saying that a person shouldn't buy anything but Apple products. The Apple II proved that. There are many companies dedicated to providing quality add-ons with reliable customer support. And you can bet dimes to dollars that almost anything without the Apple name on it will be less expensive. But unless you've really done your homework on a system, and you know enough about it to get around any problems that *might* come up with alien boards and plugs and things, start off with all name brand merchandise.

Get the minimum system you can learn with, and, as you grow more aware of the needs of the system and the reliability of it and everyone else, then flesh it out. If you're more comfortable, at that point, with Apple products, no problem. Likewise, if you're not, no problem either.

So much for the hardware of it. I don't want to get overly precise and start talking about bits and bytes and bank selecting memory and 6502s versus 6502Bs. These are all things that I think are nice to know if you want to know them, but I don't think anyone should *have* to know them just to use a machine.

Basically Speaking

There are very few things I really *must* know about a computer when I buy it. How to turn it on and where what plugs into what, of course, are top on the list. But the single thing that I can't wait to find in the manual is how to clear the screen. Only problem is, it's the one thing that is never on the first page of the instructions. Unfortunately, after I see all the nonsense and the copyright notice, it's usually the first thing I want to do.

Let's go down the list of a few Apple III commands that might be helpful immediately.

HOME—clears the screen and leaves the cursor in the upper left corner (the "home" position for the Apple III).

VPOS—sets the vertical line on

which you want the cursor or anything printed. It can be from 1 to 24.

HPOS—is like VPOS, but it sets the horizontal position of the cursor or printed character. It can be from 1 to 80.

GET—is a way of taking single characters typed at the keyboard into a program. When you use GET, the program stops until you press a key.

INPUT—use this to type in a name or some piece of information longer than one character. It stops the program and takes in all the keys you press until it sees the return key.

PRINT—is a "show-off" command. Once you've done something you're proud enough of to let someone see, PRINT it.

IF. . .THEN—checks for a certain condition and, IF the program finds it, THEN it will go and do something about it.

Basic with Dignity

I wouldn't want to guess how many programmers use those commands (or ones just like them) for the bulk of all the programming there is. Let's write a small program (see the program listing) and find out how they work. If you happen to have an Apple III, put away VisiCalc for a few moments and try typing it in.

Like almost all forms of Basic, Business Basic uses line numbers to keep track of what's supposed to happen where and when in a program. You'll use them when you write your powerful program (c'mon, *everybody* says his program is powerful, why shouldn't you?).

But hold on there, that program uses more than a few simple commands! All right, settle in there. It's not that difficult. HOME clears the screen; VPOS = 5 starts whatever you want to do on the 5th vertical position from the top of the viewing screen; HPOS = 1 fixes things exactly at the 1st horizontal position of line 5.

INPUT might have given you a little trouble, but it's logical enough. Rather than have a user just sit there looking at a computer that requires a response, Basic lets you include a prompt to tell people what the program is looking for (otherwise you'd have to use a PRINT statement, then the INPUT).

A Question of Marks

The only demand Basic makes is that prompts be enclosed in quotes and followed by either a comma, if you don't want to automatically print a question mark when the program is run, or a semicolon, if you do. (F'rinstance, as it's written, line 40 will look like this when the program is run:

What is your name?

THE LEMON™ SOURS SURGES



Our crop-The Lemon™, The Lime™, and The Orange™ are designed to eliminate undetected submicrosecond overvoltage transients from electrical circuits. Commonly referred to as "spikes", or "glitches", these transients can cause hardware and software damage to unprotected circuits.

Today's electronic products are often microprocessor controlled - mini and micro computers, televisions, video cassette recorders - to name a few. Each of these products is sensitive to fluctuations in electrical power lines. Power switching devices such as refrigerators coming on and off or air conditioners starting up can be responsible for a momentary surge or spike of electricity in a circuit. Even your local utility stepping-up transformers to add power at peak load times or an electrical storm passing through can trigger surges. Such surges can cause equipment to falter at times, not to work at peak performance or fail completely. An entire data base can be lost.

Now you can prevent this from happening to you with an AC Surge Protector from Electronic Protection Devices. Each Protector is a solid state clamping device with 6 outlets utilizing modern high speed semiconductor technology. Using our Protectors is as simple as plugging it into any standard three wire duplex outlet then plugging what needs protection into it. Each Protector exceeds the IEEE 587-1980 Guide for Surge Voltages in Low Voltage AC Power Circuits.

When you compare the cost of computer hardware, software and your time with the price of a Protector (from $59.95 to $139.95), you'll want to sour your surges with one of the AC Surge Protectors from EPD, which are available through your local dealer.

Electronic Protection Devices
5 Central Avenue
Waltham, Massachusetts 02154

In Massachusetts Call:
(617) 891-6602

Outside Massachusetts Call:
1-800-343-1813

Dealer Inquiries Invited

Circle 25 on Reader Service card.

but, if you had used a semicolon, it would look like this:

What is your name??

Double question marks might be good sometimes, but it's nice to be able to program them out.)

At the end of that line, you've created a variable, called NAME$. I had no special reason for calling it that, except that it will help to remind you later on what the variable stands for without having to go through half a dozen lines to figure it out. The III allows long variable names, so you can be very descriptive. The $ says that this is a string variable, and may contain a mixture of alphabetic and/or numeric characters.

"You wouldn't want to send the program off into the twilight zone."

What Goes In Must Come Out

Lines 50, 60 and 70 are no problem—you've seen them before. Line 80, though, is an experience. Here you're trying to make the program a little friendlier by addressing the user by name. PRINT will, of course, take the section in quotes and print it on the screen. The semicolon at the end of the line instructs Basic to stop there and use that point as the starting point for whatever is printed next. In this case, it's the user's name, found in NAME$.

The possibilities with PRINT are a comma, a semicolon or nothing. A comma tells Basic to print whatever comes next in the first available tab position that occurs *after whatever was just printed* (most computers have this feature, with "tabs" marked at the first screen column, the ninth, and after every eight spaces until the end of the line). Since nothing occurs after NAME$, Basic will print a carriage return, to bring the next print position back to the beginning of the line, and a line feed, to bring it down to the next line so you don't print over what you've just written.

. . . and In Again

That takes care of lines 90 and 100 as well. What's next? Ah yes, line 110. I've already told you about GET, and here it is in action. Once here, the computer waits with the utmost patience until you press one of the keys on the keyboard. Then it springs into action! Quickly, it finds out what key was pressed, assigns its value to the variable you've told it to expect, and then, faster than a speeding nanosecond, it moves on to the next line of the program. Not bad for a three-letter instruction!

By now I think you realize that KEY$ is just a variable name made up for convenience, just as NAME$ was.

After pressing the key, you go through 120 and 130, convey the social amenities with 140, and just take a moment to make sure that the user did press the key you asked for.

Branching with Conditionals

(A serious subtitle, take note!)

We're checking to see IF the value assigned to KEY$ was either a capital Y or a lowercase y (either is acceptable; we don't want to put too much strain on the user). If it wasn't, THEN GOTO line 200.

"Huh?", you say. It's understandable, since I did just sort of throw that one at you. Use the less than (<) and greater than (>) signs to judge for nonequalities. Maybe it will help to read it in English instead of Basic:

If key-string does not equal capital Y and keystring does not equal lowercase y then go to line 200.

That's a little better, isn't it? Some things remain unexplained, like how we got the GOTO. In the case of Apple III Business Basic, making a valid line number (now, c'mon, you wouldn't want to try to send the program off into the twilight zone, would you?) the object of the THEN is the same as telling the program to goto that line number if the preset conditions are met. Some versions of Basic require the GOTO; this one doesn't.

Okay, but what's a GOTO? It's an instruction to Basic that tells it to go to a line number; you can see how this works in line 170 (but that's getting a little ahead of yourself).

If, in line 150, you find that KEY$ *does* equal Y or y, the program ignores the THEN command and (in Business Basic as well as some others) it ignores anything and everything else that follows within the same line number. There is an exception, but for now just keep it in mind. That sends it right down to line 160, which shows that even computers have feelings, and line 170 after that, where the program goes to line 300.

If, oh horror of horrors, KEY$ *does not* equal Y or y, then, according to the instructions, it goes to line 200, where, being the arrogant, self-sufficient beast it is, the computer berates the poor human for not following directions.

In either case, both *branches* of the program wind up at line 300. The END command isn't required by Business Basic, but since you'd probably like to wrap this up at some point, use it. You can place it wherever you won't accidentally run into it, and send the program to it when appropriate.

The Tunnel at the End of the Light

Do you see how easy it is? But don't reach for that Bud yet (or, if you're like me, that egg cream). I'd like to show you just one more thing.

Each of the lines you've written is called a Basic statement. Business Basic, having been written by the Great Programmers, does not waste space, so you don't have to use separate line numbers for each statement. You can combine those that work together. Lines 10, 20 and 30 could be written:

```
10 HOME:VPOS=5:HPOS=1
```

as could lines 50, 60 and 70:

```
50 HOME:VPOS=5:HPOS=1
```

and lines 120 and 130:

```
120 VPOS=10:HPOS=1
```

(Even the GET KEY$ could be part of it, but for clarity it isn't). You must, however, separate the statements with colons, and preserve some special considerations for the IF...THEN condition.

You could change things a little so the *conditional* goes like this:

```
150 IF KEY$="Y" THEN PRINT"You got the key.":GOTO 300
```

This would work fine unless KEY$ did not equal Y. At that point, it would ignore the GOTO 300 and continue on to the next line. You'd have to eliminate lines 160 and 170, just in case. In this particular program it's not critical, but if you had programmed in a number of choices, it could get tricky. The Great Programmers have provided an out for us in case we get into trouble: the ELSE statement. You can use it like this:

```
150 IF KEY$="Y" THEN PRINT"Great!":
ELSE PRINT"Not so great"
```

That's the only statement Business Basic will recognize following an untrue IF...THEN conditional. You *must* include the colon (this is an oddity of Business Basic—most other versions don't need it).

You've got enough to play with (did I mention that we were playing? Well, we are, because if we were doing work, it wouldn't be any fun). In fact, you've probably got enough to get yourself into not-so-serious trouble.

Next time we might take a look at all those strange messages you're going to get when you try to key that program in (don't get me wrong, the program's okay, but if you've never done it before, you might be surprised).

Ciao bene, AppleAmerica.... ■

edited by Linda Stephenson

A Musical Note

Note Trespassing is a musical game from Notable Software, PO Box 1556, Philadelphia, PA 19105. High-resolution graphics are used to display manuscript-quality notes which move across the musical staff. The object is to eliminate all of the notes, by matching them with their letter names, before they can reach the end of the row. As each note is identified, its actual pitch is played through the Apple speaker (no additional hardware is required). Options include: choice of clefs (treble, bass, soprano, alto, tenor), three levels of difficulty, and muted response. Game paddles required. Price is $25. Reader Service number 120.

Shape Tables Tamed

A combination utility disk called Apple Mechanic is available from Beagle Bros. Software, 4315 Sierra Vista, San Diego, CA 92103. Two excellent shape writing programs are on the disk, including a character editor for creating full 96-character shape fonts. Three listable demo programs are included too. The Byte Zap program can display any sector of a disk in hex, decimal or ASCII format. Any byte can then be edited under Cursor Control and written back to the disk. Apple Mechanic is "unlocked," so you can make backup copies. Price is $29.50. Reader Service number 121.



Physical Science

Three new programs for high schools or introductory college courses are offered by Vernier Software, 2920 SW 89th St., Portland, OR 97225. Graphical Analysis plots graphs in high-resolution graphics and includes several features to make it especially useful in science classes; it costs $24.95. Precision Timer allows the computer to be used as a versatile lab timer, displaying times to a tenth of a millisecond; price is $39.95. It is designed to use with photogates. Ray Tracer draws ray diagrams to illustrate the principles of geometrical optics; price is $24.95. Reader Service number 123.

School Software

The Academic Secretary keeps track of all the student data that schools need to maintain. Names, addresses, parent or guardian information, emergency phone numbers, homeroom, locker number with key code or combination, book and materials check-out inventory, special funding grant information and more are kept in the system. The program comes with a file transfer program for transferring data from CMA's Grading Systems, Class Scheduling System and the Attendance program. The Academic Secretary costs $249.95. Mid-Term Report automatically prepares down-slips, warning slips and progress reports at the end of grading periods. Roster data can be transferred from the Grading System, or entered manually. Price for Mid-Term Report is $249.95.

Charles Mann & Associates, 55722 Santa Fe Trail, Yucca Valley, CA 92284. Reader Service number 124.

Structure Your Apple III

Modula-2, a superset of Pascal data and statement structures and standard data types, now runs on the Apple III. Benefits include low-level machine access coupled with library modules that provide all useful system facilities.

For more information contact Volition Systems, PO Box 1236, Del Mar, CA 92014. Reader Service number 125.

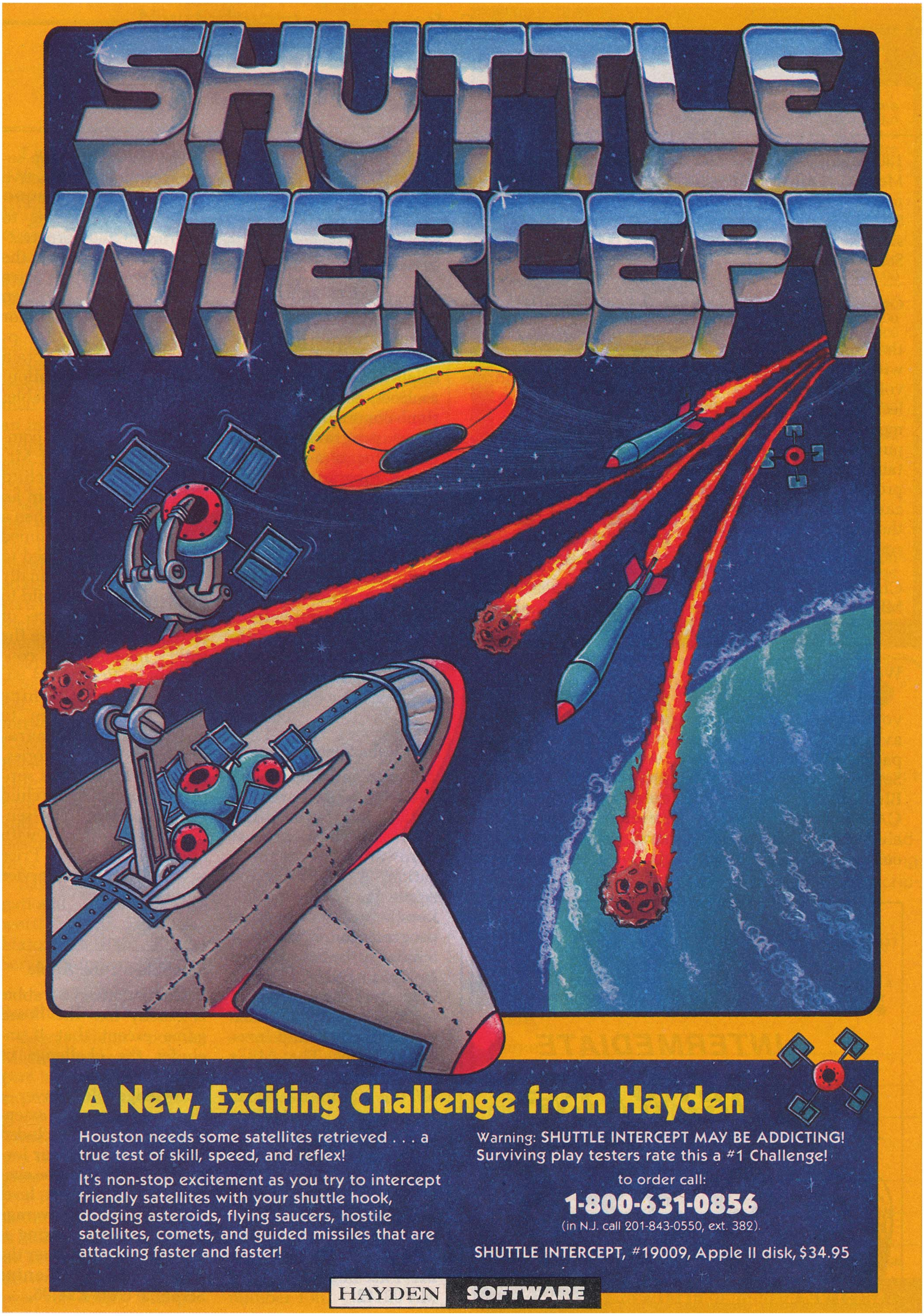
Business Projections

A financial projection and forecasting system for the Apple III is available from Management Control Concept Inc., 124 St. Mary's St., Boston, MA 02215. ProForma uses more than 700 preprogrammed conditional commands, and can calculate virtually unlimited possible variations. The program allows business-people to prepare budgets, plans, forecasts and analyses. Reader Service number 126.

Apple III Medical Package

Charles Mann & Associates, 55722 Santa Fe Trail, Yucca Valley, CA 92284, offers a medical applications program to handle appointment scheduling, private patient billing, Superbill preparation, general patient recall and third party claims form preparation for the medium-sized medical office. Up to 13,000 appointments can be made, printed onto labels, canceled, searched for, or printed onto organized daily schedules.

The system requires an Apple III with SOS and Business Basic, at least 128K of RAM (256K for the hard disk version) with two to four floppy disk drives, or an Apple Profile or Corvus hard disk, and a



Circle 61 on Reader Service card.

130-column printer. The Medical Office Management package costs $1595.95; hard disk version is $1695.95. Reader Service number 127.

One of a Kind

Know Your Apple tells the story of your Apple with animated graphics, voice, music and text. Select any lesson from the menu page, and the computer will teach and entertain the newcomer. The program comes boxed in a cardboard miniature of the Apple computer; it costs $34.95.

Muse Software, 347 N. Charles St., Baltimore, MD 21201. Reader Service number 128.

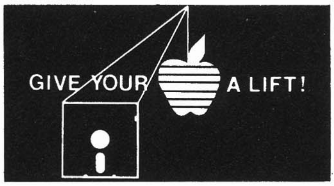
Word Games

Fast-paced competitive vocabulary games are available in two new packages from Don't Ask Software, 2265 Westwood Blvd., B-150, Los Angeles, CA 90064. WordRace includes three levels of difficulty, designed for youngsters, teenagers and word sharks. Each turn starts with a word, six possible definitions, and a counter set to 600 points. As time passes the number of points decreases—when you pick the correct definition you get the points left, but, if you guess wrong, you lose the number of points left and the clock keeps ticking.

An accessory disk, Intermediate WordRace, contains a fourth level of difficulty; it's aimed at adults or teens but is less challenging than the top level in WordRace. Reader Service number 122.

Programming Utility

Menu Generator is a software package for developing menu programs on the Apple II. It's easy to use and requires no programming to develop professional menus. Menu programs are defined by entering data on one screen form. Menu Generator will compile this data, write a menu program in Basic, and store it on a specified disk. The menu programs feature formatted and centered displays, error processing routines, and disk operation commands to run other programs. Menu Generator is a complete menu program development system; it can store data for up to 150 menus. Its database has a unique file selection method for adding, editing and deleting menus. Price is $39.95.



Crane Software Inc., 16835 Algonquin, Suite 611, Huntington Beach, CA 92649. Reader Service number 131.



Mill Enhancement

Stellation Two, PO Box 2342, Santa Barbara, CA 93120, has begun shipment of their newest hardware/software application, A.S.A.P. for the Apple II. The system allows a variety of popular programs to use The Mill. Similar to the Pascal Speed Up System, A.S.A.P. works with software intended for the Pascal Run Time environment, and executes under The Mill's 6809 coprocessor instead of the standard 6502. Price is $295. Reader Service number 129.

Graphic Games

Educational games that entertain are offered by Spinnaker Software Corp., 215 First St., Cambridge, MA 02142.

The Snooper Troop series contains interactive mysteries; players are private detectives trying to determine who perpetrated the crime and what the motive was. Players drive around town, question suspects, search houses for clues, and use the Snoopnet computer to obtain background information. Two cases are available; each costs $44.95.

Facemaker is a computerized spin-off of Mr. Potatohead. The child constructs a face from an inventory of parts. Then a succession of programmed commands can be used to animate the face. Finally the game asks the child to repeat the sequence of faces, to help improve the child's memory and concentration. Price is $34.95.

The Story Machine helps children learn to write sentences, paragraphs and simple stories. A list of 40 words is supplied from which the child selects his text. The sentence is illustrated with animated graphics. Price is $34.95. Reader Service number 130.

Scrabble for the Apple

Computerized Scrabble is exactly like the board game except that it includes the challenge of playing against a very knowledgeable player—the computer. An extensive dictionary is included in the program. Four levels of difficulty can be chosen—at the highest level the computer will average 300 points. All scoring is kept automatically on the screen. Little Genius Scrabble costs £24.95.

Little Genius Ltd., Suite 504, Albany House, 324 Regent St., London W1R5AA, England. Reader Service number 132.

Plain Language Computing

The Savvy Personal Language System, using a pattern recognition processor, lets users work in their own words. Savvy is an information processing system that can "learn" to operate in the same pattern as the person using it. It will do what it's asked in the user's own language. For instance, an accounts payable program could be adapted to respond to a statement or command such as, "Give me all my invoices for July," or, "I need my July shipments."

Savvy consists of a manual, firmware board and software disks; it provides General Ledger, Accounts Receivable, Accounts Payable, Payroll, Mailing List, Document Writer and Inventory Control programs. Savvy package price is $950.

Savvy Marketing International, 100 South Ellsworth St., 9th Floor, San Mateo, CA 94401. Reader Service number 133.

Another Adventure

Scott Adams' popular Mission Impossible Adventure is now available with superb high-resolution graphics for the Apple II. Time is of the essence as you race the clock to complete your mission—for if you fail, the world's first automated nuclear reactor is doomed. The graphics in Mission Impossible are drawn from a palette of over 100 colors. The program also supports Votrax Type 'N Talk voice synthesizer, to give an adventure that talks. Price is $29.95.

Adventure International, Dept. G, Box 3435, Longwood, FL 32750. Reader Service number 134.

Soft Graphics

The Poor Man's Graphics Tablet has a virtually unlimited palette of colors that can be applied in more than 59 textures. A unique marking feature lets the user trace transparencies laid over the screen: Just mark a few points of the figure and the program traces the figure for you. Full shape table functions are included as well as full manipulation of shapes and pictures. All manipulation and drawing are done through the Apple keyboard. Price is $49.95.

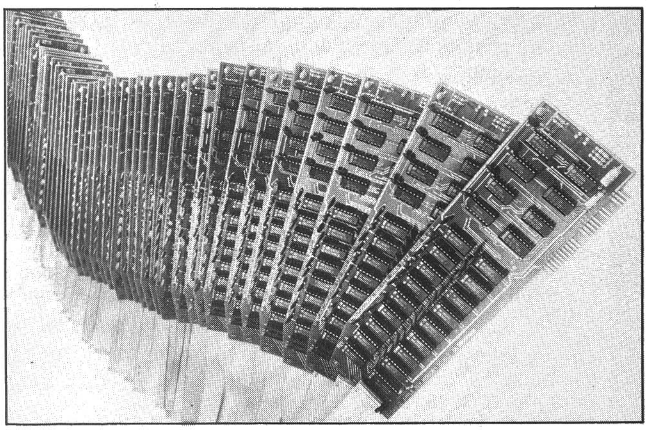
Rainbow Computing, 19517 Business Center Drive, Northridge, CA 91324. Reader Service number 135.

Sound Entertainment

By plugging any audio input into the Apple II cassette port, a user can activate real-time full color graphics, synchronized to music. Kaleido-Sound comes with four kaleidoscopic patterns, each with variable color schemes. The user can alter the colors used and vary the dynamics of the musically-generated graphics. It's a multimedia entertainment experience, and costs $39.95.

Passport Designs Inc., 116 North Cabrillo Highway, Half Moon Bay, CA 94019. Reader Service number 136.

edited by Linda Stephenson



RAM cards from U-Microcomputers will expand your Apple II memory.

More Memory

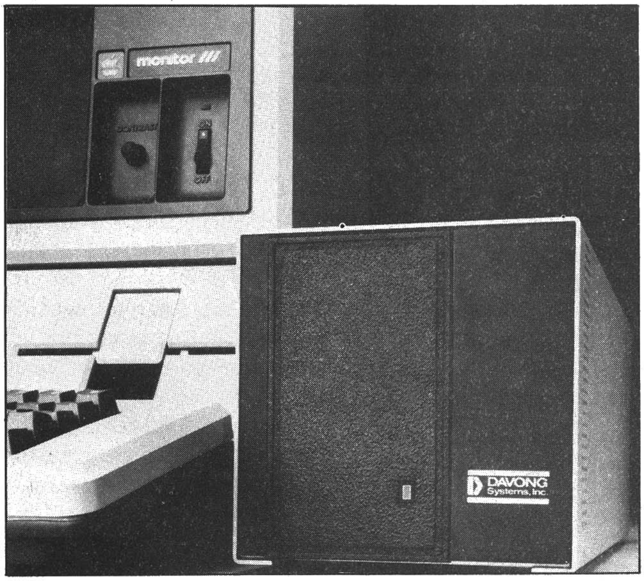
A range of RAM cards offering 32K, 64K and 128K bytes of memory is offered by U-Microcomputers Ltd., 300 Broad St., Stamford, CT 06901. These cards complement the available 16K card to provide up to 320K.

Software from U-Microcomputers for use with these cards includes VisiCalc expansion software and RAM disk emulation software. Apple II users can now run much bigger VisiCalc models. Reader Service number 101.

Logo Newsletter

The National Logo Exchange is a monthly newsletter that provides practical suggestions for using Logo in the classroom. Published September through May, it serves as a forum for the exchange of ideas, philosophies and techniques of teaching and using Logo. Columns by well-known professional educators are featured each month, as well as practical articles submitted by subscribers. This excellent reference costs $25 per year to U.S. addresses, $30 elsewhere.

The National Logo Exchange, Box 5341, Charlottesville, VA 22905, attention Wilma Goff. Reader Service number 109.



Davong Systems' hard disk for the Apple III.

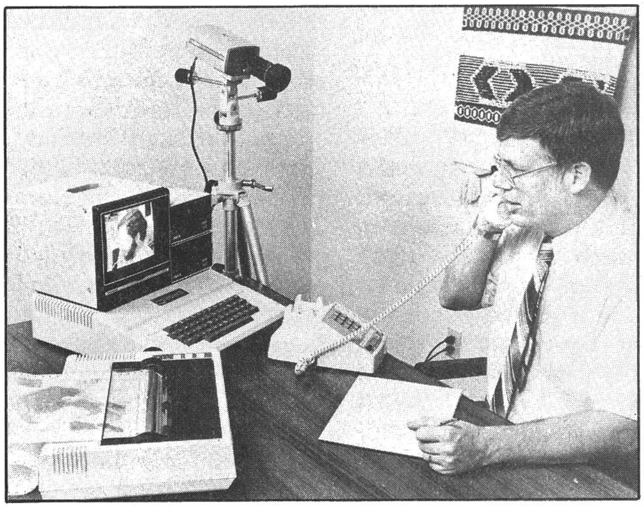
Three for III

Apple III owners can expand their machines with 5, 10 or 15 megabytes of hard disk memory. Davong Systems' line of hard disk expansion systems includes the DSI-A306, featuring 5 megabytes formatted (6M unformatted) for $1995; the DSI-A312, with 10M (12M unformatted) for $2495; and the DSI-A319, with 15M (19M unformatted) for $2995. Up to four hard disks can be daisy chained by a single controller.

Davong Systems Inc., 1061 Terra Bella Ave., Mountain View, CA 94043. Reader Service number 107.

Photo Artistry

The PhotoCaster photography system permits color or black-and-white shots taken with a standard TV camera to be displayed, processed, and transmitted over the phone lines. With PhotoCaster you can add titles and graphics to photos, create special video effects, enhance images, retrieve and store photos on disk and design slide shows. A built-in high-resolution graphics dump routine provides "prints." Black-and-white photos are processed with a resolution of 128 × 128 pixels and 16 levels of gray; shades of gray are represented on the monitor by dot dithering. Color photography yields eight-color photos with 16 saturation levels. Color photos are taken using a three-frame RGB sequence with color filters. Price of the PhotoCaster model PC-100 is $499.95. Model PC-101, which in-



The PhotoCaster system from Commsoft.

HI-LIGHT YOUR GRAPHICS!

Bring your Apple® graphics programs out of the dark ages with the help of Avant-Garde Creations' enlightening **HI-RES SECRETS** programmer's utility series! These two information-packed systems reveal everything you need to know about state-of-the-art Apple® hi-res graphics programming and more. Both are stand-alone packages, but together they make an unbeatable combination!

NOW IN THE SPOTLIGHT:

HI-RES SECRETS GRAPHICS APPLICATIONS SYSTEM (G.A.S.) is a complete utility that takes you step by step through the answers to all of your specific Apple® graphics application questions including: how to turn fair BASIC programs into good BASIC programs, translate BASIC programs into machine language programs, how to make business graphics, architectural and electronic designs, 3-D designs, marketable quality arcade and adventure games and more. **G.A.S.** also contains an exciting new super-fast color-filling Palette program that has 140 gorgeous colors, 160 different patterns, 4 separate fill algorithms and the program even allows you to fill on both black and white backgrounds! **HI-RES SECRETS GRAPHICS APPLICATIONS SYSTEM** includes 3 unprotected disks plus an extensive manual at a special introductory price of only $75.00. Available separately, **HI-RES ARCHITECTURAL DESIGN** and **HI-RES ELECTRONIC DESIGN** retail for $29.95 each.

HAVE YOU SEEN THE LIGHT?

HI-RES SECRETS is Avant-Garde Creations' original best-selling programmer's graphics utility. From block, vector and hplot shapes to animation, type font, music and color-fill programs, **HI-RES SECRETS** reveals all! From novice to expert, this indispensible graphics utility system is a must for every programmer's library. **HI-RES SECRETS** contains 4 disks with a 263-page manual and retails for $125.00.

No permission or royalties are required for the use of **HI-RES SECRETS** or **G.A.S.** routines in your own programs. All 3 systems run on Apple II 48K DOS 3.3 and are available at your local dealer or you may order direct from Avant-Garde Creations.

Apple® is a registered trademark of Apple Computer Inc.

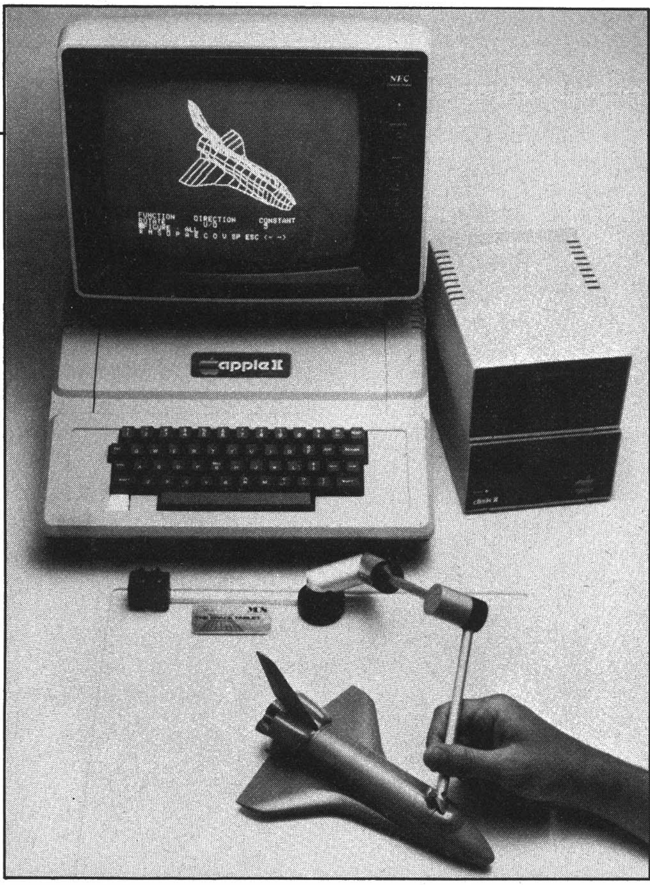


AVANT-GARDE CREATIONS

Circle 39 on Reader Service card.

P.O. BOX 30160 • EUGENE, OR 97403 • (503) 345-3043

BUSINESS, EDUCATION, UTILITY AND ENTERTAINMENT SOFTWARE



The Space Tablet spatial digitizer from Micro Control Systems.

cludes a Panasonic camera and RGB color filter accessory, costs $749.95.

Commsoft Inc., 2452 Embarcadero Way, Palo Alto, CA 94303. Reader Service number 105.

Create 3-D Graphics

Now Apple owners can record the x, y and z coordinates of any three-dimensional object. The Space Tablet, from Micro Control Systems Inc., 143 Tunnel Road, Vernon, CT 06066, comes complete with Space Graphics or Penguin Software's Complete Graphics System II. Space Graphics incorporates many features of large CAD systems; a single motion of the tablet arm pointer can pull points and lines to new locations in three-dimensional space. The Space Tablet is priced under $600. Reader Service number 103.

Voice Synthesizer

The V100A Interactive Voice Synthesizer allows users to call or be called by a computer and to receive voice messages from a program or local database. This single-board product also lets the user respond with Touch-Tone signals to computer queries for modifying the program flow, updating the database and so on. No modem is required. The synthesizer plugs directly into the phone system.

The master disk provided with the unit contains 300 common words and a selection of terms used by specific industries. Customized vocabularies on disk can be purchased from the manufacturer. The V100A for the Apple II costs $395.

Vynet Corp., 2405 Qume Drive, San Jose, CA 95131. Reader Service number 102.

Printer Interfaces

Two new products are offered by Advanced Logic Systems, 1195 E. Arques Ave., Sunnyvale, CA 94086. The Dispatcher is a serial RS-232 interface used to connect the Apple II to a letter quality printer, a modem, an external terminal or another computer. The PrinterMate is a parallel interface used to connect the Apple II to any Centronics-type dot matrix printer. The PrinterMate costs $99; The Dispatcher is $139. Reader Service number 106.

Apple-Compatible

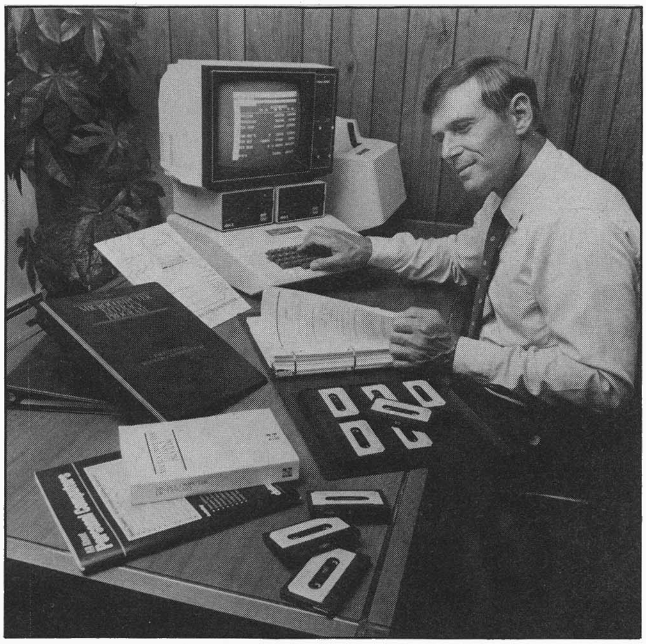
The Franklin Ace 10 floppy disk system is based on Shugart SA 400 technology—it incorporates a direct drive stepping motor actuator using a spiral cam with a U-groove positive detent. The new drive was designed to read and write any Apple II-compatible disk. Price for a single drive and controller is $579; a second drive can be added for $479. Reader Service number 114.

Fetal Monitoring

Now the Apple II can be used to perform obstetrical ultrasound examinations in the hospital, clinic or office. Obus provides enhanced fetal evaluation, speeds up the interpretation and reporting process and reduces transcription needs. A specially designed graphics table is used to enter information on fetal survey, presentation, and placental and amniotic fluid characteristics. In this way, measurements of fetal parameters are rapidly obtained from the ultrasound images. Calculations of gestational age and weight, comparison with anticipated gesta-



Franklin Ace 10 disk drive.



The McGraw-Hill at-home seminar takes the mystery out of the microcomputer.

tional age and weight, and assessment of fetal growth are then automatically performed. A final report is generated and then stored on disk. Because the computer is off-line to the ultrasound equipment, procedures will not be delayed if computer problems arise.

For information write to Trinity Computing Systems Inc., 1020 Holcombe Blvd., Suite 408, Houston, TX 77030. Reader Service number 115.

Executive Computing

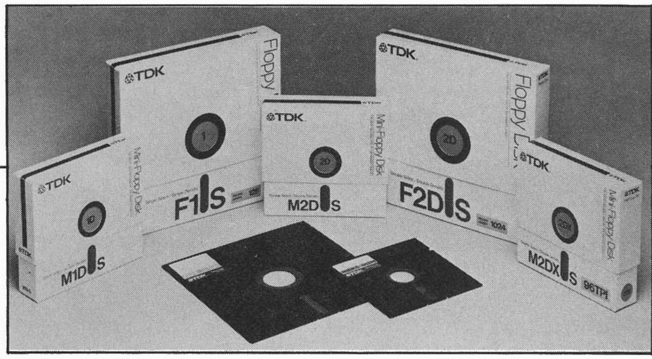
A new "Microcomputer Literacy Program for Executives, Managers and Professionals" is being introduced by the McGraw-Hill Book Company. The program combines text, audio cassettes and supplementary materials designed to demystify the office micro. Price is $129.95.

McGraw-Hill Continuing Education Center, 3939 Wisconsin Ave. NW, Washington, DC 20016. Reader Service number 110.

Classy Graphics

The Digisolve graphics processor plugs into your Apple II and provides a monochrome graphics display of 512×512 pixels. The on-board vector graphics processor draws lines and characters at up to 1,500,000 pixels per second, freeing the Apple's processor for other tasks. The processor is designed for use with assembly language and Basic. Price is £399.

Digisolve Ltd., Lambson Group, Aire & Calder Works, Cinder Lane, Castleford, West Yorks WF10 1LU. Reader Service number 108.



TDK's new line of premium-quality floppy disks.

Floppy Disks

TDK Electronics Corp., 12 Harbor Park Drive, Port Washington, NY 11050, offers a line of durable floppy disks with an extended warranty. Double-density 5¼-inch disks come in double-sided and single-sided versions. The eight-inch disks are available as single-sided/single-density or double-sided/double-density. Prices start at $5.50. Reader Service number 111.

Soft Slot

People who have "everything" for their Apple may find that the one thing they lack is an empty slot. The Soft 8 card creates one more slot, and it's accessible under program control.

The Slot 8 plugs into slot 7 and provides slots 7 and 8. You can PR#8, IN#8 and catalog slot 8. The disk supplied with the Soft 8 lets you modify standard Apple DOS to recognize the added slot. Price is $84.95.

Legend Industries Ltd., PO Box 112, Pontiac, MI 48056. Reader Service number 104.

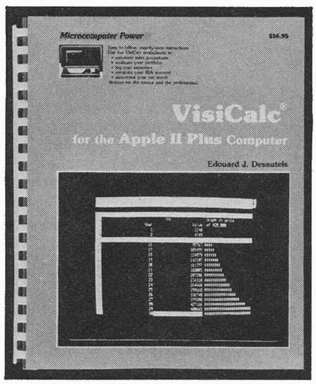
Bubble Memory

The Apple II Bubble Memory Module provides 128K bytes of nonvolatile memory. It's ideal for use as a fast pseudo disk in conjunction with a floppy disk to hold files that are frequently accessed for programs or data. It will save time and increase the life of the floppy disk. The stand-alone capabilities provided by the on-board boot PROM give the user a non-volatile mass memory that is more rugged than a floppy and is completely portable. Price is $875.

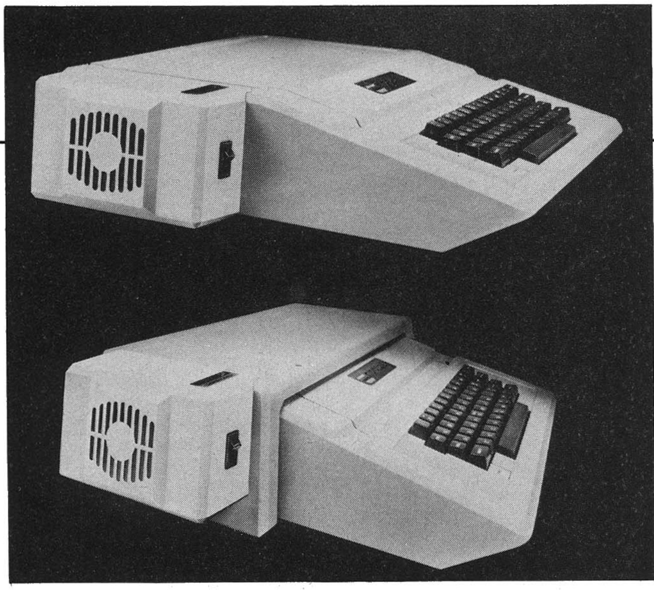
MPC Peripherals Corp., 9424 Chesapeake Drive, San Diego, CA 92123. Reader Service number 100.

VisiCalc Tutorial

VisiCalc for the Apple II Plus is a practical guidebook for serious spreadsheeters. Edouard J. Desautels, a computer science professor at the University of Wisconsin, shares his expertise and



The VisiCalc guidebook from Wm. Brown Company.



The redesigned System Saver, from Kensington Microware, fits the Apple II with or without the Apple monitor stand.

VisiCalc programming knowledge in a simple workbook format. The book provides clear, Apple-specific examples to help the programmer learn faster and work more efficiently. It costs $16.95, and is available from Wm. C. Brown Company Publishers, 2460 Kerper Blvd., Dubuque, IA 52001. Reader Service number 116.

A Cool Breeze for Apples

A newly designed System Saver for the Apple II is available from Kensington Microware Ltd., 300 East 54th St., Suite 3L, New York, NY 10022. System Saver protects the computer with efficient line surge suppression, quiet cooling, a strong 16 amp front-mounted power switch and two rear-mounted outlets controlled from the front switch and protected by the surge suppression circuitry. Price is $89.95.

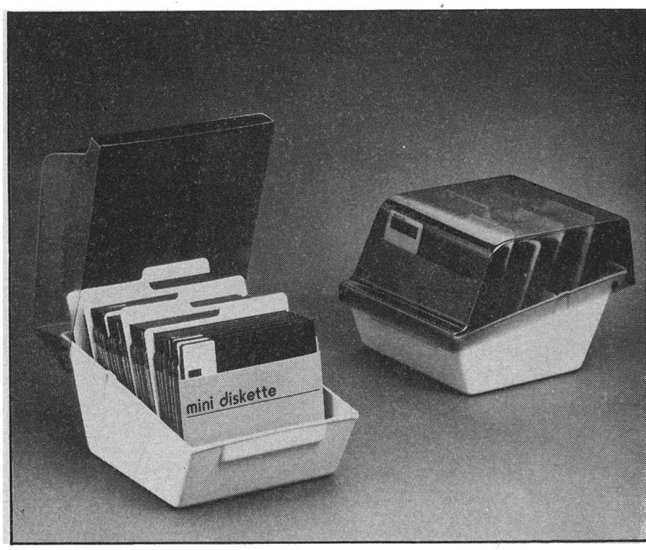
Smart Storage

Plastic disk trays from Ring King Visibles Inc., 215 West Second St., PO Box 599, Muscatine, IA 52761, help organize your disk library for easy access to programs. Each tray holds 50 disks, and comes with five index dividers for your convenience. Foam cushioning helps reduce contact between tray and media. A smoked plastic lid permits identification of the tray's contents without opening. Price is $35.95. Reader Service number 112.

Free Catalogs

Monument Computer Service, PO Box 603, Joshua Tree, CA 92252, offers a free catalog of educational and personal programs for Apple II and Apple III computers. Applications for classroom management and educational games are featured. Reader Service number 117.

A free catalog listing 30 software programs is available from the Institute for Computers in Jewish Life, 845 N. Michigan, Suite 843, Chicago, IL 60611. The programs include Hebrew language and Jewish history, music and customs. Educational video games such as "Jewish I.Q. Baseball," "Purimaze" and "The Game of the Maccabees" are of special interest. Reader Service number 118.



Disk storage trays from Ring King Visibles.

TURBOCHARGE YOUR APPLE*

With The Quentin 500 Winchester Disk Drive



Introductory Prices

5 MB	$1995
10 MB	$2195
15 MB	$2395
20 MB	$2595

When you're ready to go the distance, the Quentin 500 is the high density, high speed Winchester subsystem that can fuel your Apple into a first-class powerhouse.

It's Deep. Store and retrieve 5, 10, 15 even 20 megabytes of formatted data on a single drive. With 20 megabytes of juice, you've got the equivalent of 140 conventional floppy disks, 4800 full pages of text, or all the financial files of a $50 million business.

It's Fast. The Quentin 500 accelerates from 0 to *Read* in 70 MS. And the DMA data transfer rate zips along at 5 MBits per second.

It's Accurate. The Quentin 500 moves out in front with the remarkable new Disk Drive Self Test Diagnostic System. Flick a switch and the 500 runs through its paces checking all Winchester operating functions, and verifying the integrity of the drive and the data. While the controller's full 32 bit Error Correction Code overrides all data field errors.

It's Reliable. Whether you're on-line all day, or driving in short, fast sprints, the Quentin 500 is strictly high performance. MTBF 10,000 power-on hours, with no preventive maintenance required.

Every Quentin 500 is delivered completely assembled, and is fully tested on Apple operating systems. And the Q-500 has a full one-year factory warranty.

It's Compatible and Apple-Beige. Plug it into your favorite Apple, and shift into full power. Software support packages provided to ensure complete compatibility with DOS, CP/M, Pascal and protected software.

And It's Very Affordable. The Quentin 500 is the first premium quality Winchester subsystem offered at a price all businessmen, professional corporations and serious computer users can afford.

Count On Quentin For Quality. Quentin Research has been designing and building mass storage subsystems for corporate users for five years. That's quite a few times around the track.

We've put those five years to good use by developing state-of-the-art disk technology for business and personal use, and by building a reputation for reliable products and stable management.

The Hotline to our Product Support Manager is open every week day for information on our full product line. When you're ready to expand your computer's capabilities, turbocharge with the Quentin 500. It's an information powerhouse.

QUENTIN RESEARCH, INC.
19355 Business Center Drive
Northridge, CA 91324
(213) 701-1006 • Telex 910-493-2174

To Order Check, money order, Visa or Mastercard number. Calif. residents add 6% sales tax. Authorized club discounts available. Call or write for Quentin 500 Specifications and Features.

* Apple is a registered trademark of Apple Computer, Inc.

Circle 30 on Reader Service card.